

**INVERSE PROBLEMS IN
GEOPHYSICS
GEOS 567**

A Set of Lecture Notes

by

**Professors Randall M. Richardson and George Zandt
Department of Geosciences
University of Arizona
Tucson, Arizona 85721**

Revised and Updated Summer 2003

TABLE OF CONTENTS

PREFACEv

CHAPTER 1: INTRODUCTION 1

 1.1 Inverse Theory: What It Is and What It Does 1

 1.2 Useful Definitions 2

 1.3 Possible Goals of an Inverse Analysis 3

 1.4 Nomenclature 4

 1.5 Examples of Forward Problems 7

 1.5.1 Example 1: Fitting a Straight Line 7

 1.5.2 Example 2: Fitting a Parabola 8

 1.5.3 Example 3: Acoustic Tomography 9

 1.5.4 Example 4: Seismic Tomography 10

 1.5.5 Example 5: Convolution 10

 1.6 Final Comments 11

CHAPTER 2: REVIEW OF LINEAR ALGEBRA AND STATISTICS 12

 2.1 Introduction 12

 2.2 Matrices and Linear Transformations..... 12

 2.2.1 Review of Matrix Manipulations 12

 2.2.2 Matrix Transformations 15

 2.2.3 Matrices and Vector Spaces 19

 2.2 Probability and Statistics 20

 2.3.1 Introduction 20

 2.3.2 Definitions, Part 1 20

 2.3.3 Some Comments on Applications to Inverse Theory 23

 2.3.4 Definitions, Part 2 24

CHAPTER 3: INVERSE METHODS BASED ON LENGTH 28

 3.1 Introduction 28

 3.2 Data Error and Model Parameter Vectors 28

 3.3 Measures of Length 28

 3.4 Minimizing the Misfit: Least Squares 30

 3.4.1 Least Squares Problem for a Straight Line 30

 3.4.2 Derivation of the General Least Squares Solution 33

 3.4.3 Two Examples of Least Squares Problems 35

 3.4.4 Four-Parameter Tomography Problem 37

 3.5 Determinancy of Least Squares Problems..... 38

 3.5.1 Introduction 38

 3.5.2 Even-Determined Problems: $M = N$ 39

 3.5.3 Overdetermined Problems: Typically, $N > M$ 39

 3.5.4 Underdetermined Problems: Typically $M > N$ 39

 3.6 Minimum Length Solution 40

 3.6.1 Background Information 40

 3.6.2 Lagrange Multipliers 41

 3.6.3 Application to the Purely Underdetermined Problem 44

3.6.4	Comparison of Least Squares and Minimum Length Solutions	46
3.6.5	Example of Minimum Length Problem	46
3.7	Weighted Measures of Length	47
3.7.1	Introduction	47
3.7.2	Weighted Least Squares	47
3.7.3	Weighted Minimum Length	50
3.7.4	Weighted Damped Least Squares	52
3.8	A Priori Information and Constraints	53
3.8.1	Introduction	53
3.8.2	A First Approach to Including Constraints	54
3.8.3	A Second Approach to Including Constraints	56
3.8.4	Example From Seismic Receiver Functions	59
3.9	Variance of the Model Parameters	60
3.9.1	Introduction	60
3.9.2	Application to Least Squares	60
3.9.3	Application to the Minimum Length Problem	61
3.9.4	Geometrical Interpretation of Variance	61
CHAPTER 4: LINEARIZATION OF NONLINEAR PROBLEMS		65
4.1	Introduction	65
4.2	Linearization of Nonlinear Problems	65
4.3	General Procedure for Nonlinear Problems	68
4.4	Three Examples	68
4.4.1	A Linear Example	68
4.4.2	A Nonlinear Example	70
4.4.3	Nonlinear Straight-Line Example	75
4.5	Creeping vs Jumping (<i>Shaw and Orcutt, 1985</i>)	79
CHAPTER 5: THE EIGENVALUE PROBLEM		82
5.1	Introduction	82
5.2	The Eigenvalue Problem for Square ($M \times M$) Matrix \mathbf{A}	82
5.2.1	Background	82
5.2.2	How Many Eigenvalues, Eigenvectors?	83
5.2.3	The Eigenvalue Problem in Matrix Notation	84
5.2.4	Summarizing the Eigenvalue Problem for \mathbf{A}	87
5.3	Geometrical Interpretation of the Eigenvalue Problem for Symmetric \mathbf{A}	87
5.3.1	Introduction	87
5.3.2	Geometrical Interpretation	88
5.3.3	Coordinate System Rotation	92
5.3.4	Summarizing Points	93
5.4	Decomposition Theorem for Square \mathbf{A}	94
5.4.1	The Eigenvalue Problem for \mathbf{A}^T	94
5.4.2	Eigenvectors for \mathbf{A}^T	94
5.4.3	Decomposition Theorem for Square Matrices	95
5.4.4	Finding the Inverse \mathbf{A}^{-1} for the $M \times M$ Matrix \mathbf{A}	101
5.4.5	What Happens When There Are Zero Eigenvalues?	102
5.4.6	Some Notes on the Properties of \mathbf{S}_P and \mathbf{R}_P	105
5.5	Eigenvector Structure of \mathbf{m}_{LS}	106
5.5.1	Square Symmetric \mathbf{A} Matrix With Nonzero Eigenvalues	106
5.5.2	The Case of Zero Eigenvalues	108
5.5.3	Simple Tomography Problem Revisited	109

CHAPTER 6:	SINGULAR-VALUE DECOMPOSITION (SVD)	113
6.1	Introduction	113
6.2	Formation of a New Matrix \mathbf{B}	113
6.2.1	Formulating the Eigenvalue Problem With \mathbf{G}	111
6.2.2	The Role of \mathbf{G}^T as an Operator	114
6.3	The Eigenvalue Problem for \mathbf{B}	115
6.3.1	Properties of \mathbf{B}	115
6.3.2	Partitioning \mathbf{W}	115
6.4	Solving the Shifted Eigenvalue Problem	116
6.4.1	The Eigenvalue Problem for $\mathbf{G}^T\mathbf{G}$	116
6.4.2	The Eigenvalue Problem for $\mathbf{G}\mathbf{G}^T$	118
6.5	How Many η_i Are There, Anyway??	119
6.5.1	Introducing P , the Number of Nonzero Pairs $(+\eta_i, -\eta_i)$	119
6.5.2	Finding the Eigenvector Associated with $-\eta_i$	120
6.5.3	No New Information From the $-\eta_i$ System	121
6.5.4	What About the Zero Eigenvalues η_i 's, $i = 2(P + 1), \dots, N + M$?	121
6.5.5	How Big is P ?	122
6.6	Introducing Singular Values	123
6.6.1	Introduction	123
6.6.2	Definition of the Singular Value	124
6.6.3	Definition of Λ , the Singular-Value Matrix	124
6.7	Derivation of the Fundamental Decomposition Theorem for General \mathbf{G} ($N \times M, N \neq M$)	126
6.8	Singular-Value Decomposition (SVD)	127
6.8.1	Derivation of Singular-Value Decomposition	127
6.8.2	Rewriting the Shifted Eigenvalue Problem	129
6.8.3	Summarizing SVD	129
6.9	Mechanics of Singular-Value Decomposition	131
6.10	Implications of Singular-Value Decomposition	132
6.10.1	Relationships Between \mathbf{U} , \mathbf{U}_P , and \mathbf{U}_0	132
6.10.2	Relationships Between \mathbf{V} , \mathbf{V}_P , and \mathbf{V}_0	132
6.10.3	Graphic Representation of \mathbf{U} , \mathbf{U}_P , \mathbf{U}_0 , \mathbf{V} , \mathbf{V}_P , and \mathbf{V}_0 Spaces	133
6.11	Classification of $\mathbf{d} = \mathbf{G}\mathbf{m}$ Based on P , M , and N	134
6.11.1	Introduction	132
6.11.2	Class I: $P = M = N$	135
6.11.3	Class II: $P = M < N$	135
6.11.4	Class III: $P = N < M$	136
6.11.5	Class IV: $P < \min(N, M)$	137
CHAPTER 7:	THE GENERALIZED INVERSE AND MEASURES OF QUALITY	138
7.1	Introduction	138
7.2	The Generalized Inverse Operator \mathbf{G}_g^{-1}	140
7.2.1	Background Information	140
7.2.2	Class I: $P = N = M$	140
7.2.3	Class II: $P = M < N$	141
7.2.4	Class III: $P = N < M$	148
7.2.5	Class IV: $P < \min(N, M)$	151

7.3	Measures of Quality for the Generalized Inverse	153
7.3.1	Introduction	153
7.3.2	The Model Resolution Matrix \mathbf{R}	153
7.3.3	The Data Resolution Matrix \mathbf{N}	156
7.3.4	The Unit (Model) Covariance Matrix $[\text{cov}_u \mathbf{m}]$	159
7.3.5	Combining \mathbf{R} , \mathbf{N} , $[\text{cov}_u \mathbf{m}]$	161
7.3.6	An Illustrative Example	164
7.4	Quantifying the Quality of \mathbf{R} , \mathbf{N} , and $[\text{cov}_u \mathbf{m}]$	166
7.4.1	Introduction	166
7.4.2	Classes of Problems	166
7.4.3	Effect of the Generalized Inverse Operator \mathbf{G}_g^{-1}	167
7.5	Resolution Versus Stability	169
7.5.1	Introduction	169
7.5.2	\mathbf{R} , \mathbf{N} , and $[\text{cov}_u \mathbf{m}]$ for Nonlinear Problems	171
CHAPTER 8: VARIATIONS OF THE GENERALIZED INVERSE		176
8.1	Linear Transformations	176
8.1.1	Analysis of the Generalized Inverse Operator \mathbf{G}_g^{-1}	176
8.1.2	\mathbf{G}_g^{-1} Operating on a Data Vector \mathbf{d}	178
8.1.3	Mapping Between Model and Data Space: An Example	179
8.2	Including Prior Information, or the Weighted Generalized Inverse	181
8.2.1	Mathematical Background	181
8.2.2	Coordinate System Transformation of Data and Model Parameter Vectors	183
8.2.3	The Maximum Likelihood Inverse Operator, Resolution, and Model Covariance	185
8.2.4	Effect on Model- and Data-Space Eigenvectors	187
8.2.5	An Example	189
8.3	Damped Least Squares and the Stochastic Inverse	195
8.3.1	Introduction	195
8.3.2	The Stochastic Inverse	195
8.3.3	Damped Least Squares	199
8.4	Ridge Regression	203
8.4.1	Mathematical Background	203
8.4.2	The Ridge Regression Operator	204
8.4.3	An Example of Ridge Regression Analysis	205
8.5	Maximum Likelihood	210
8.5.1	Background	210
8.5.2	The General Case.....	212
CHAPTER 9: CONTINUOUS INVERSE THEORY AND OTHER APPROACHES.....		216
9.1	Introduction	216
9.2	The Backus–Gilbert Approach	217
9.3	Neural Networks	225
9.4	The Radon Transform and Tomography (Approach 1).....	227
9.4.1	Introduction	227
9.4.2	Interpretation of Tomography Using the Radon Transform	230
9.4.3	Slant-Stacking as a Radon Transform (following <i>Claerbout, 1985</i>)	231
9.5	A Review of the Radon Transform (Approach 2)	235
9.6	Alternative Approach to Tomography	238

PREFACE

This set of lecture notes has its origin in a nearly incomprehensible course in inverse theory that I took as a first-semester graduate student at MIT. My goal, as a teacher and in these notes, is to present inverse theory in such a way that it is not only comprehensible but useful.

Inverse theory, loosely defined, is the fine art of inferring as much as possible about a problem from all available information. Information takes both the traditional form of data, as well as the relationship between actual and predicted data. In a nuts-and-bolt definition, it is one (some would argue the best!) way to find and assess the quality of a solution to some (mathematical) problem of interest.

Inverse theory has two main branches dealing with *discrete* and *continuous* problems, respectively. This text concentrates on the discrete case, covering enough material for a single-semester course. A background in linear algebra, probability and statistics, and computer programming will make the material much more accessible. Review material is provided on the first two topics in Chapter 2.

This text could stand alone. However, it was written to complement and extend the material covered in the required text for the course, which deals more completely with some areas. Furthermore, these notes make numerous references to sections in the required text. Besides, the required text is, by far, the best textbook on the subject and should be a part of the library of anyone interested in inverse theory. The required text is:

Geophysical Data Analysis: Discrete Inverse Theory (Revised Edition)
by William Menke, Academic Press, 1989.

The course format is largely lecture. We may, from time to time, read articles from the literature and work in a seminar format. I will try and schedule a couple of guest lectures in applications. Be forewarned. There is a lot of homework for this course. They are occasionally very time consuming. I make every effort to avoid pure algebraic nightmares, but my general philosophy is summarized below:

I hear, and I forget.
I see, and I remember.
I do, and I understand.
– Chinese Proverb

I try to have you do a “simple” problem by hand before turning you loose on the computer, where all realistic problems must be solved. You will also have access to existing code and a computer account on my SPARC10 workstation. You may use and modify the code for some of the homework and for the term project. The term project is an essential part of the learning process and, I hope, will help you tie the course work together. Grading for this course will be as follows:

60%	Homework
30%	Term Project
10%	Class Participation

Good luck, and may you find the trade-off between stability and resolution less traumatic than most, on average.

Randy Richardson and George Zandt

CHAPTER 1: INTRODUCTION

1.1 Inverse Theory: What It Is and What It Does

Inverse theory, at least as I choose to define it, is the fine art of estimating model parameters from data. It requires a knowledge of the forward model capable of predicting data if the model parameters were, in fact, already known. Anyone who attempts to solve a problem in the sciences is probably using inverse theory, whether or not he or she is aware of it. Inverse theory, however, is capable (at least when properly applied) of doing much more than just estimating model parameters. It can be used to estimate the “quality” of the predicted model parameters. It can be used to determine which model parameters, or which combinations of model parameters, are best determined. It can be used to determine which data are most important in constraining the estimated model parameters. It can determine the effects of noisy data on the stability of the solution. Furthermore, it can help in experimental design by determining where, what kind, and how precise data must be to determine model parameters.

Inverse theory is, however, inherently mathematical and as such does have its limitations. It is best suited to estimating the numerical values of, and perhaps some statistics about, model parameters for some *known* or *assumed* mathematical model. It is less well suited to provide the fundamental mathematics or physics of the model itself. I like the example Albert Tarantola gives in the introduction of his classic book¹ on inverse theory. He says, “. . . you can always measure the captain’s age (for instance by picking his passport), but there are few chances for this measurement to carry much information on the number of masts of the boat.” You must have a good idea of the applicable forward model in order to take advantage of inverse theory. Sooner or later, however, most practitioners become rather fanatical about the benefits of a particular approach to inverse theory. Consider the following as an example of how, or how not, to apply inverse theory. The existence or nonexistence of a God is an interesting question. Inverse theory, however, is poorly suited to address this question. However, if one assumes that there is a God and that She makes angels of a certain size, then inverse theory might well be appropriate to determine the number of angels that could fit on the head of a pin. Now, who said practitioners of inverse theory tend toward the fanatical?

In the rest of this chapter, I will give some useful definitions of terms that will come up time and again in inverse theory, and give some examples, mostly from Menke’s book, of how to set up forward problems in an attempt to clearly identify model parameters from data.

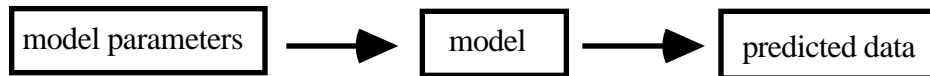
¹*Inverse Problem Theory*, by Albert Tarantola, Elsevier Scientific Publishing Company, 1987.

1.2 Useful Definitions

Let us begin with some definitions of things like *forward* and *inverse* theory, *models* and *model parameters*, *data*, etc.

Forward Theory: The (mathematical) process of predicting data based on some physical or mathematical model with a given set of model parameters (and perhaps some other appropriate information, such as geometry, etc.).

Schematically, one might represent this as follows:



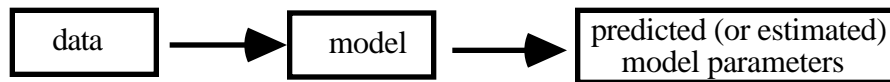
As an example, consider the two-way vertical travel time t of a seismic wave through M layers of thickness d_i and velocity v_i . Then t is given by

$$t = 2 \sum_{i=1}^M \frac{d_i}{v_i} \quad (1.1)$$

The forward problem consists of predicting data (travel time) based on a (mathematical) model of how seismic waves travel. Suppose that for some reason thickness was known for each layer (perhaps from drilling). Then only the M velocities would be considered model parameters. One would obtain a particular travel time t for each set of model parameters one chooses.

Inverse Theory: The (mathematical) process of predicting (or estimating) the numerical values (and associated statistics) of a set of model parameters of an assumed model based on a set of data or observations.

Schematically, one might represent this as follows:



As an example, one might invert the travel time t above to determine the layer velocities. Note that one needs to know the (mathematical) model relating travel time to layer thickness and velocity information. Inverse theory should not be expected to provide the model itself.

Model: The model is the (mathematical) relationship between model parameters (and other auxiliary information, such as the layer thickness information in the previous example) and the data. It may be linear or nonlinear, etc.

Model Parameters: The model parameters are the numerical quantities, or unknowns, that one is attempting to estimate. The choice of model parameters is usually problem dependent, and quite often arbitrary. For example, in the case of travel times cited earlier, layer thickness is *not*

considered a model parameter, while layer velocity is. There is nothing sacred about these choices. As a further example, one might choose to cast the previous example in terms of slowness s_i , where:

$$s_i = 1 / v_i \quad (1.2)$$

Travel time t is a nonlinear function of layer velocities but a linear function of layer slowness. As you might expect, it is much easier to solve linear than nonlinear inverse problems. A more serious problem, however, is that linear and nonlinear formulations may result in different estimates of velocity if the data contain any noise. The point I am trying to impress on you now is that there is quite a bit of freedom in the way model parameters are chosen, and it can affect the answers you get!

Data: Data are simply the observations or measurements one makes in an attempt to constrain the solution of some problem of interest. Travel time in the example above is an example of data. There are, of course, many other examples of data.

Some examples of inverse problems (mostly from Menke) follow:

- Medical tomography
- Earthquake location
- Earthquake moment tensor inversion
- Earth structure from surface or body wave inversion
- Plate velocities (kinematics)
- Image enhancement
- Curve fitting
- Satellite navigation
- Factor analysis

1.3 Possible Goals of an Inverse Analysis

Now let us turn our attention to some of the possible goals of an inverse analysis. These might include:

1. Estimates of a set of model parameters (obvious).
2. Bounds on the range of acceptable model parameters.
3. Estimates of the formal uncertainties in the model parameters.
4. How sensitive is the solution to noise (or small changes) in the data?
5. Where, and what kind, of data are best suited to determine a set of model parameters?
6. Is the fit between predicted and observed data adequate?
7. Is a more complicated (i.e., more model parameters) model significantly better than a more simple model?

Not all of these are completely independent goals. It is important to realize, as early as possible, that there is much more to inverse theory than simply a set of estimated model parameters. Also, it is important to realize that there is very often not a single “correct” answer. Unlike a mathematical inverse, which either exists or does not exist, there are many possible approximate inverses. These may give different answers. Part of the goal of an inverse analysis is to determine if the “answer” you have obtained is reasonable, valid, acceptable, etc. This takes experience, of course, but you have begun the process.

Before going on with how to formulate the mathematical methods of inverse theory, I should mention that there are two basic branches of inverse theory. In the first, the model parameters and data are discrete quantities. In the second, they are continuous functions. An example of the first might occur with the model parameters we seek being given by the moments of inertia of the planets:

$$\text{model parameters} = I_1, I_2, I_3, \dots, I_{10} \quad (1.3)$$

and the data being given by the perturbations in the orbital periods of satellites:

$$\text{data} = T_1, T_2, T_3, \dots, T_N \quad (1.4)$$

An example of a continuous function type of problem might be given by velocity as a function of depth:

$$\text{“model parameters”} = v(z) \quad (1.5)$$

and the data given by a seismogram of ground motion

$$\text{“data”} = d(t) \quad (1.6)$$

Separate strategies have been developed for discrete and continuous inverse theory. There is, of course, a fair bit of overlap between the two. In addition, it is often possible to approximate continuous functions with a discrete set of values. There are potential problems (aliasing, for example) with this approach, but it often makes otherwise intractable problems tractable. Menke’s book deals exclusively with the discrete case. This course will certainly emphasize discrete inverse theory, but I will also give you a little of the continuous inverse theory at the end of the semester.

1.4 Nomenclature

Now let us introduce some nomenclature. In these notes, vectors will be denoted by boldface lowercase letters, and matrices will be denoted by boldface uppercase letters.

Suppose one makes N measurements in a particular experiment. We are trying to determine the values of M model parameters. Our nomenclature for data and model parameters will be

$$\text{data: } \mathbf{d} = [d_1, d_2, d_3, \dots, d_N]^T \quad (1.7)$$

$$\text{model parameters: } \mathbf{m} = [m_1, m_2, m_3, \dots, m_M]^T \quad (1.8)$$

where \mathbf{d} and \mathbf{m} are N and M dimensional column vectors, respectively, and T denotes transpose.

The model, or relationship between \mathbf{d} and \mathbf{m} , can have many forms. These can generally be classified as either *explicit* or *implicit*, and either *linear* or *nonlinear*.

Explicit means that the data and model parameters *can* be separated onto different sides of the equal sign. For example,

$$d_1 = 2m_1 + 4m_2 \quad (1.9)$$

and

$$d_1 = 2m_1 + 4m_1^2 m_2 \quad (1.10)$$

are two explicit equations.

Implicit means that the data *cannot* be separated on one side of an equal sign with model parameters on the other side. For example,

$$d_1(m_1 + m_2) = 0 \quad (1.11)$$

and

$$d_1(m_1 + m_1^2 m_2) = 0 \quad (1.12)$$

are two implicit equations. In each example above, the first represents a *linear* relationship between the data and model parameters, and the second represents a *nonlinear* relationship.

In this course we will deal exclusively with *explicit* type equations, and predominantly with *linear* relationships. Then, the *explicit linear* case takes the form

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

where \mathbf{d} is an N -dimensional data vector, \mathbf{m} is an M -dimensional model parameter vector, and \mathbf{G} is an $N \times M$ matrix containing only constant coefficients.

The matrix \mathbf{G} is sometimes called the *kernel* or *data kernel* or even the Green's function because of the analogy with the continuous function case:

$$\mathbf{d}(x) = \int \mathbf{G}(x, t)\mathbf{m}(t) dt \quad (1.14)$$

Consider the following discrete case example with two observations ($N = 2$) and three model parameters ($M = 3$):

$$\begin{aligned} d_1 &= 2m_1 + 0m_2 - 4m_3 \\ d_2 &= m_1 + 2m_2 + 3m_3 \end{aligned} \quad (1.15)$$

which may be written as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -4 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (1.16)$$

or simply

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

where

$$\mathbf{d} = [d_1, d_2]^T$$

$$\mathbf{m} = [m_1, m_2, m_3]^T$$

and

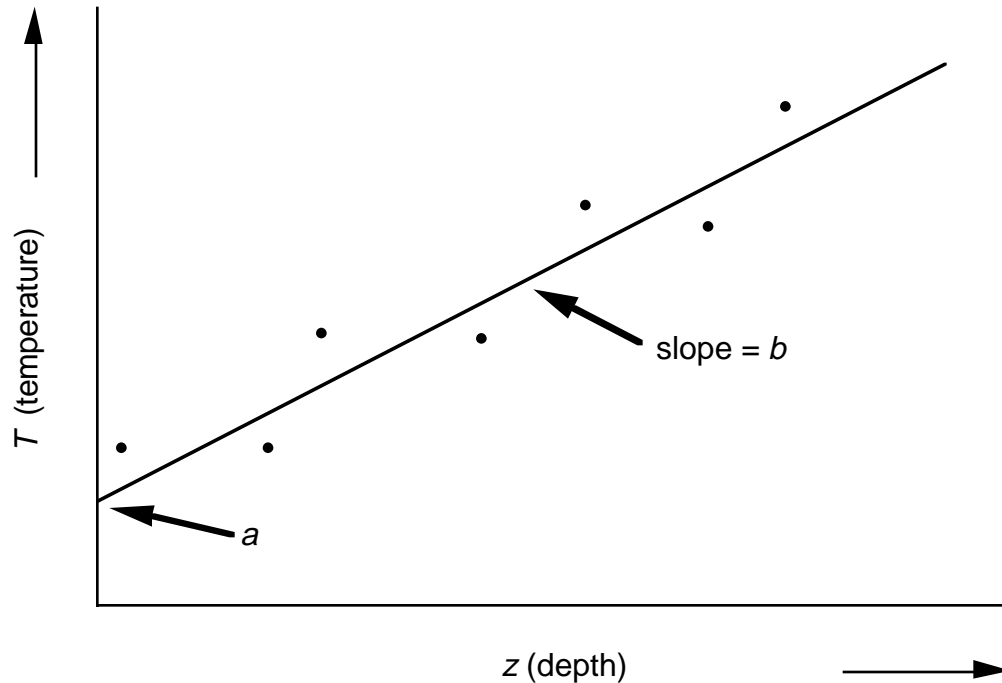
$$\mathbf{G} = \begin{bmatrix} 2 & 0 & -4 \\ 1 & 2 & 3 \end{bmatrix} \quad (1.17)$$

Then \mathbf{d} and \mathbf{m} are 2×1 and 3×1 column vectors, respectively, and \mathbf{G} is a 2×3 matrix with constant coefficients.

On the following pages I will give some examples of how forward problems are set up using matrix notation. See pages 10–16 of Menke for these and other examples.

1.5 Examples of Forward Problems

1.5.1 Example 1: Fitting a Straight Line (See Page 10 of Menke)



Suppose that N temperature measurements T_i are made at depths z_i in the earth. The data are then a vector \mathbf{d} of N measurements of temperature, where $\mathbf{d} = [T_1, T_2, T_3, \dots, T_N]^T$. The depths z_i are not data. Instead, they provide some auxiliary information that describes the geometry of the experiment. This distinction will be further clarified below.

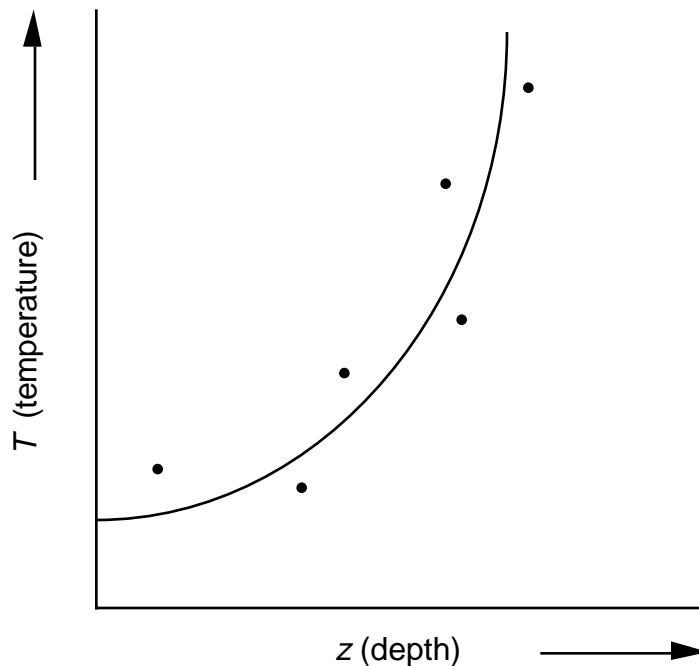
Suppose that we assume a model in which temperature is a linear function of depth: $T = a + bz$. The intercept a and slope b then form the two model parameters of the problem, $\mathbf{m} = [a, b]^T$. According to the model, each temperature observation must satisfy $T = a + bz$:

$$\begin{aligned} T_1 &= a + bz_1 \\ T_2 &= a + bz_2 \\ &\vdots \\ T_N &= a + bz_N \end{aligned}$$

These equations can be arranged as the matrix equation $\mathbf{Gm} = \mathbf{d}$:

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

1.5.2 Example 2: Fitting a Parabola (See Page 11 of Menke)



If the model in example 1 is changed to assume a quadratic variation of temperature with depth of the form $T = a + bz + cz^2$, then a new model parameter is added to the problem, $\mathbf{m} = [a, b, c]^T$. The number of model parameters is now $M = 3$. The data are supposed to satisfy

$$\begin{aligned} T_1 &= a + bz_1 + cz_1^2 \\ T_2 &= a + bz_2 + cz_2^2 \\ &\vdots \\ T_N &= a + bz_N + cz_N^2 \end{aligned}$$

These equations can be arranged into the matrix equation

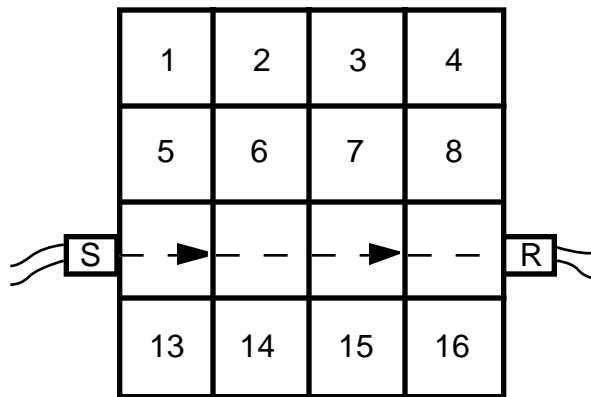
$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ \vdots & \vdots & \vdots \\ 1 & z_N & z_N^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

This matrix equation has the explicit linear form $\mathbf{Gm} = \mathbf{d}$. Note that, although the equation is linear in the data and model parameters, it is not linear in the auxiliary variable z .

The equation has a very similar form to the equation of the previous example, which brings out one of the underlying reasons for employing matrix notation: it can often emphasize similarities between superficially different problems.

1.5.3 Example 3: Acoustic Tomography (See Pages 12–13 of Menke)

Suppose that a wall is assembled from a rectangular array of bricks (Figure 1.1 from Menke, below) and that each brick is composed of a different type of clay. If the acoustic velocities of the different clays differ, one might attempt to distinguish the different kinds of bricks by measuring the travel time of sound across the various rows and columns of bricks, in the wall. The data in this problem are $N = 8$ measurements of travel times, $\mathbf{d} = [T_1, T_2, T_3, \dots, T_8]^T$. The model assumes that each brick is composed of a uniform material and that the travel time of sound across each brick is proportional to the width and height of the brick. The proportionality factor is the brick's *slowness* s_i , thus giving $M = 16$ model parameters, $\mathbf{m} = [s_1, s_2, s_3, \dots, s_{16}]^T$, where the ordering is according to the numbering scheme of the figure as



The travel time of acoustic waves (dashed lines) through the rows and columns of a square array of bricks is measured with the acoustic source S and receiver R placed on the edges of the square. The inverse problem is to infer the acoustic properties of the bricks (which are assumed to be homogeneous).

$$\begin{array}{ll}
 \text{row 1:} & T_1 = hs_1 + hs_2 + hs_3 + hs_4 \\
 \text{row 2:} & T_2 = hs_5 + hs_6 + hs_7 + hs_8 \\
 \vdots & \vdots \\
 \text{column 4:} & T_8 = hs_4 + hs_8 + hs_{12} + hs_{16}
 \end{array}$$

and the matrix equation is

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_8 \end{bmatrix} = h \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{16} \end{bmatrix}$$

Here the bricks are assumed to be of width *and* height h .

1.5.4 Example 4: Seismic Tomography

An example of the impact of inverse methods in the geosciences: Northern California

- A large amount of data is available, much of it redundant.
- Patterns in the data can be interpreted qualitatively.
- Inversion results quantify the patterns.
- Perhaps, more importantly, inverse methods provide quantitative information on the resolution, standard error, and "goodness of fit."
- We cannot overemphasize the "impact" of colorful graphics, for both good and bad.
- Inverse theory is not a magic bullet. Bad data will still give bad results, and, interpretation of even good results requires breadth of understanding in the field.
- Inverse theory does provide quantitative information on how well the model is "determined," importance of data, and model errors.
- Another example: improvements in "imaging" subduction zones.

1.5.5 Example 5: Convolution

Convolution is widely significant as a physical concept and offers an advantageous starting point for many theoretical developments. One way to think about convolution is that it describes the action of an observing instrument when it takes a weighted mean of some physical quantity over a narrow range of some variable. All physical observations are limited in this way, and for this reason alone convolution is ubiquitous (paraphrased from Bracewell, *The Fourier Transform and Its Applications*, 1964). It is widely used in time series analysis as well to represent physical processes.

The convolution of two functions $f(x)$ and $g(x)$ represented as $f(x)*g(x)$ is

$$\int_{-\infty}^{\infty} f(u) g(x-u) du \quad (1.18)$$

For discrete finite functions with common sampling intervals, the convolution is

$$h_k = \sum_{i=0}^m f_i g_{k-i} \quad 0 < k < m+n \quad (1.19)$$

A FORTRAN computer program for convolution would look something like:

```

      L=M+N-1
      DO 10 I=1,L
10    H(I)=0
      DO 20 I=1,M
      DO 20 J=1,N
20    H(I+J-1)=H(I+J-1)+G(I)*F(J)

```

Convolution may also be written using matrix notation as

$$\begin{bmatrix} f_1 & 0 & \cdot & \cdot & \cdot & 0 \\ f_2 & f_1 & \cdot & \cdot & \cdot & 0 \\ \cdot & f_2 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & f_1 \\ f_n & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & f_n & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & f_n \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ \cdot \\ g_m \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ \cdot \\ h_{n+m-1} \end{bmatrix} \quad (1.20)$$

In the matrix form, we recognize our familiar equation $\mathbf{Gm} = \mathbf{d}$ (ignoring the confusing notation differences between fields, when, for example, g_1 above would be m_1), and we can define deconvolution as the inverse problem of finding $\mathbf{m} = \mathbf{G}^{-1}\mathbf{d}$. Alternatively, we can also reformulate the problem as $\mathbf{G}^T\mathbf{Gm} = \mathbf{G}^T\mathbf{d}$ and find the solution as $\mathbf{m} = [\mathbf{G}^T]^{-1} [\mathbf{G}^T\mathbf{d}]$.

1.6 Final Comments

The purpose of the previous examples has been to help you formulate forward problems in matrix notation. It helps you to clearly differentiate model parameters from other information needed to calculate “predicted” data. It also helps you separate data from everything else. Getting the forward problem set up in matrix notation is essential before you can invert the system.

The logical next step is to take the forward problem given by

$$\mathbf{d} = \mathbf{Gm} \quad (1.13)$$

and invert it for an estimate of the model parameters \mathbf{m}^{est} as

$$\mathbf{m}^{\text{est}} = \mathbf{G}^{\text{“inverse”}} \mathbf{d} \quad (1.17)$$

We will spend a lot of effort determining just what $\mathbf{G}^{\text{“inverse”}}$ means when the inverse does not exist in the mathematical sense of

$$\mathbf{G}\mathbf{G}^{\text{“inverse”}} = \mathbf{G}^{\text{“inverse”}}\mathbf{G} = \mathbf{I} \quad (1.18)$$

where \mathbf{I} is the identity matrix.

The next order of business, however, is to shift our attention to a review of the basics of *matrices and linear algebra* as well as *probability and statistics* in order to take full advantage of the power of inverse theory.

CHAPTER 2: REVIEW OF LINEAR ALGEBRA AND STATISTICS

2.1 Introduction

In discrete inverse methods, matrices and linear transformations play fundamental roles. So do probability and statistics. This review chapter, then, is divided into two parts. In the first, we will begin by reviewing the basics of matrix manipulations. Then we will introduce some special types of matrices (Hermitian, orthogonal and semiorthogonal). Finally, we will look at matrices as linear transformations that can operate on vectors of one dimension and return a vector of another dimension. In the second section, we will review some elementary probability and statistics, with emphasis on Gaussian statistics. The material in the first section will be particularly useful in later chapters when we cover eigenvalue problems, and methods based on the length of vectors. The material in the second section will be very useful when we consider the nature of noise in the data and when we consider the maximum likelihood inverse.

2.2 Matrices and Linear Transformations

Recall from the first chapter that, by convention, vectors will be denoted by lower case letters in boldface (i.e., the data vector \mathbf{d}), while matrices will be denoted by upper case letters in boldface (i.e., the matrix \mathbf{G}) in these notes.

2.2.1 Review of Matrix Manipulations

Matrix Multiplication

If \mathbf{A} is an $N \times M$ matrix (as in N rows by M columns), and \mathbf{B} is an $M \times L$ matrix, we write the $N \times L$ product \mathbf{C} of \mathbf{A} and \mathbf{B} , as

$$\mathbf{C} = \mathbf{AB} \quad (2.1)$$

We note that matrix multiplication is associative, that is

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}) \quad (2.2)$$

but in general is not commutative. That is, in general

$$\mathbf{AB} \neq \mathbf{BA} \quad (2.3)$$

In fact, if \mathbf{AB} exists, then the product \mathbf{BA} only exists if \mathbf{A} and \mathbf{B} are square.

In Equation (2.1) above, the ij th entry in \mathbf{C} is the product of the i th row of \mathbf{A} and the j th column of \mathbf{B} . Computationally, it is given by

$$c_{ij} = \sum_{k=1}^M a_{ik} b_{kj} \quad (2.4)$$

One way to form **C** using standard FORTRAN code would be

```

DO 300 I = 1, N
DO 300 J = 1, L
C(I,J) = 0.0
DO 300 K = 1, M
300 C(I,J) = C(I,J) + A(I,K)*B(K,J)
    
```

(2.5)

A special case of the general rule above is the multiplication of a matrix **G** ($N \times M$) and a vector **m** ($M \times 1$):

$$\begin{matrix} \mathbf{d} & = & \mathbf{G} & \mathbf{m} \\ (N \times 1) & & (N \times M) & (M \times 1) \end{matrix} \quad (1.13)$$

In terms of computation, the vector **d** is given by

$$d_i = \sum_{j=1}^M G_{ij} m_j \quad (2.6)$$

The Inverse of a Matrix

The mathematical inverse of the $M \times M$ matrix **A**, denoted \mathbf{A}^{-1} , is defined such that:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_M \quad (2.7)$$

where \mathbf{I}_M is the $M \times M$ identity matrix given by:

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \quad (2.8)$$

($M \times M$)

\mathbf{A}^{-1} is the matrix, which when either pre- or postmultiplied by **A**, returns the identity matrix. Clearly, since only square matrices can both pre- and postmultiply each other, the mathematical inverse of a matrix only exists for square matrices.

A useful theorem follows concerning the inverse of a product of matrices:

Theorem: If
$$\mathbf{A} = \mathbf{B} \mathbf{C} \mathbf{D} \quad (2.8a)$$

$$N \times N \quad N \times N \quad N \times N \quad N \times N$$

Then \mathbf{A}^{-1} , if it exists, is given by

$$\mathbf{A}^{-1} = \mathbf{D}^{-1} \mathbf{C}^{-1} \mathbf{B}^{-1} \quad (2.8b)$$

Proof:
$$\begin{aligned} \mathbf{A}(\mathbf{A}^{-1}) &= \mathbf{BCD}(\mathbf{D}^{-1}\mathbf{C}^{-1}\mathbf{B}^{-1}) \\ &= \mathbf{BC}(\mathbf{DD}^{-1})\mathbf{C}^{-1}\mathbf{B}^{-1} \\ &= \mathbf{BC} \mathbf{I} \mathbf{C}^{-1}\mathbf{B}^{-1} \\ &= \mathbf{B}(\mathbf{CC}^{-1})\mathbf{B}^{-1} \\ &= \mathbf{BB}^{-1} \\ &= \mathbf{I} \end{aligned} \quad (2.8c)$$

Similarly, $(\mathbf{A}^{-1})\mathbf{A} = \mathbf{D}^{-1}\mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{BCD} = \dots = \mathbf{I}$ (Q.E.D.)

The Transpose and Trace of a Matrix

The transpose of a matrix \mathbf{A} is written as \mathbf{A}^T and is given by

$$(\mathbf{A}^T)_{ij} = \mathbf{A}_{ji} \quad (2.9)$$

That is, you interchange rows and columns.

The transpose of a product of matrices is the product of the transposes, in reverse order. That is,

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T \quad (2.10)$$

Just about everything we do with real matrices \mathbf{A} has an analog for complex matrices. In the complex case, wherever the transpose of a matrix occurs, it is replaced by the complex conjugate transpose of the matrix, denoted $\tilde{\mathbf{A}}$. That is,

$$\text{if} \quad \mathbf{A}_{ij} = a_{ij} + b_{ij}\mathbf{i} \quad (2.11a)$$

$$\text{then} \quad \tilde{\mathbf{A}}_{ij} = c_{ij} + d_{ij}\mathbf{i} \quad (2.11b)$$

$$\text{where} \quad c_{ij} = a_{ji} \quad (2.11c)$$

and $d_{ij} = -b_{ji}$ (2.11d)

that is, $\tilde{\mathbf{A}}_{ij} = a_{ji} - b_{ji}\mathbf{i}$ (2.11e)

Finally, the trace of \mathbf{A} is given by

$$\text{trace}(\mathbf{A}) = \sum_{i=1}^M a_{ii} \quad (2.12)$$

Hermitian Matrices

A matrix \mathbf{A} is said to be Hermitian if it is equal to its complex conjugate transpose. That is, if

$$\mathbf{A} = \tilde{\mathbf{A}} \quad (2.13a)$$

If \mathbf{A} is a real matrix, this is equivalent to

$$\mathbf{A} = \mathbf{A}^T \quad (2.13b)$$

This implies that \mathbf{A} must be square. The reason that Hermitian matrices will be important is that they have only real eigenvalues. We will take advantage of this many times when we consider eigenvalue and shifted eigenvalue problems later.

2.2.2 Matrix Transformations

Linear Transformations

A matrix equation can be thought of as a linear transformation. Consider, for example, the original matrix equation:

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

where \mathbf{d} is an $N \times 1$ vector, \mathbf{m} is an $M \times 1$ vector, and \mathbf{G} is an $N \times M$ matrix. The matrix \mathbf{G} can be thought of as an operator that operates on an M -dimensional vector \mathbf{m} and returns an N -dimensional vector \mathbf{d} .

Equation (1.13) represents an explicit, linear relationship between the data and model parameters. The operator \mathbf{G} , in this case, is said to be linear because if \mathbf{m} is doubled, for example, so is \mathbf{d} . Mathematically, one says that \mathbf{G} is a linear operator if the following is true:

$$\begin{aligned} \text{If } & \mathbf{d} = \mathbf{G}\mathbf{m} \\ \text{and } & \mathbf{f} = \mathbf{G}\mathbf{r} \\ \text{then } & [\mathbf{d} + \mathbf{f}] = \mathbf{G}[\mathbf{m} + \mathbf{r}] \end{aligned} \quad (2.14)$$

In another way to look at matrix multiplications, in the by-now-familiar Equation (1.13),

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

the column vector \mathbf{d} can be thought of as a weighted sum of the *columns* of \mathbf{G} , with the weighting factors being the elements in \mathbf{m} . That is,

$$\mathbf{d} = m_1\mathbf{g}_1 + m_2\mathbf{g}_2 + \cdots + m_M\mathbf{g}_M \quad (2.15)$$

where

$$\mathbf{m} = [m_1, m_2, \dots, m_M]^T \quad (2.16a)$$

and

$$\mathbf{g}_i = [g_{1i}, g_{2i}, \dots, g_{Ni}]^T \quad (2.16b)$$

is the i th column of \mathbf{G} . Also, if $\mathbf{G}\mathbf{A} = \mathbf{B}$, then the above can be used to infer that the first column of \mathbf{B} is a weighted sum of the columns of \mathbf{G} with the elements of the first column of \mathbf{A} as weighting factors, etc. for the other columns of \mathbf{B} . Each column of \mathbf{B} is a weighted sum of the *columns* of \mathbf{G} .

Next, consider

$$\mathbf{d}^T = [\mathbf{G}\mathbf{m}]^T \quad (2.17)$$

or

$$\begin{array}{ccc} \mathbf{d}^T & = & \mathbf{m}^T \mathbf{G}^T \\ 1 \times N & & 1 \times M \quad M \times N \end{array} \quad (2.18)$$

The row vector \mathbf{d}^T is the weighted sum of the *rows* of \mathbf{G}^T , with the weighting factors again being the elements in \mathbf{m} . That is,

$$\mathbf{d}^T = m_1\mathbf{g}_1^T + m_2\mathbf{g}_2^T + \cdots + m_M\mathbf{g}_M^T \quad (2.19)$$

Extending this to

$$\mathbf{A}^T\mathbf{G}^T = \mathbf{B}^T \quad (2.20)$$

we have that each row of \mathbf{B}^T is a weighted sum of the *rows* of \mathbf{G}^T , with the weighting factors being the elements of the appropriate *row* of \mathbf{A}^T .

In a long string of matrix multiplications such as

$$\mathbf{A}\mathbf{B}\mathbf{C} = \mathbf{D} \quad (2.21)$$

each column of \mathbf{D} is a weighted sum of the *columns* of \mathbf{A} , and each row of \mathbf{D} is a weighted sum of the *rows* of \mathbf{C} .

Orthogonal Transformations

An orthogonal transformation is one that leaves the length of a vector unchanged. We can only talk about the length of a vector being unchanged if the dimension of the vector is unchanged. Thus, only square matrices may represent an orthogonal transformation.

Suppose \mathbf{L} is an orthogonal transformation. Then, if

$$\mathbf{Lx} = \mathbf{y} \tag{2.22}$$

where \mathbf{L} is $N \times N$, and \mathbf{x}, \mathbf{y} are both N -dimensional vectors. Then

$$\mathbf{x}^T \mathbf{x} = \mathbf{y}^T \mathbf{y} \tag{2.23}$$

where Equation (2.23) represents the dot product of the vectors with themselves, which is equal to the length squared of the vector. If you have ever done coordinate transformations in the past, you have dealt with an orthogonal transformation. Orthogonal transformations rotate vectors but do not change their lengths.

Properties of orthogonal transformations. There are several properties of orthogonal transformations that we will wish to use.

First, if \mathbf{L} is an $N \times N$ orthogonal transformation, then

$$\mathbf{L}^T \mathbf{L} = \mathbf{I}_N \tag{2.24}$$

This follows from

$$\begin{aligned} \mathbf{y}^T \mathbf{y} &= [\mathbf{Lx}]^T [\mathbf{Lx}] \\ &= \mathbf{x}^T \mathbf{L}^T \mathbf{Lx} \end{aligned} \tag{2.25}$$

but $\mathbf{y}^T \mathbf{y} = \mathbf{x}^T \mathbf{x}$ by Equation (2.23). Thus,

$$\mathbf{L}^T \mathbf{L} = \mathbf{I}_N \quad (\text{Q.E.D.}) \tag{2.26}$$

Second, the relationship between \mathbf{L} and its inverse is given by

$$\mathbf{L}^{-1} = \mathbf{L}^T \tag{2.27a}$$

and

$$\mathbf{L} = [\mathbf{L}^T]^{-1} \tag{2.27b}$$

These two follow directly from Equation (2.26) above.

Third, the determinant of a matrix is unchanged if it is operated upon by orthogonal transformations. Recall that the determinant of a 3×3 matrix \mathbf{A} , for example, where \mathbf{A} is given by

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (2.28)$$

is given by

$$\begin{aligned} \det(\mathbf{A}) &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) \\ &\quad - a_{12}(a_{21}a_{33} - a_{23}a_{31}) \\ &\quad + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned} \quad (2.29)$$

Thus, if \mathbf{A} is an $M \times M$ matrix, and \mathbf{L} is an orthogonal transformations, and if

$$\mathbf{A}' = (\mathbf{L})\mathbf{A}(\mathbf{L})^T \quad (2.30a)$$

it follows that

$$\det(\mathbf{A}) = \det(\mathbf{A}') \quad (2.30b)$$

Fourth, the trace of a matrix is unchanged if it is operated upon by an orthogonal transformation, where trace (\mathbf{A}) is defined as

$$\text{trace}(\mathbf{A}) = \sum_{i=1}^M a_{ii} \quad (2.30c)$$

That is, the sum of the diagonal elements of a matrix is unchanged by an orthogonal transformation. Thus,

$$\text{trace}(\mathbf{A}) = \text{trace}(\mathbf{A}') \quad (2.30d)$$

Semiorthogonal Transformations

Suppose that the linear operator \mathbf{L} is not square, but $N \times M$ ($N \neq M$). Then \mathbf{L} is said to be semiorthogonal if and only if

$$\mathbf{L}^T\mathbf{L} = \mathbf{I}_M, \quad \text{but } \mathbf{L}\mathbf{L}^T \neq \mathbf{I}_N, \quad N > M \quad (2.31)$$

or

$$\mathbf{L}\mathbf{L}^T = \mathbf{I}_N, \quad \text{but } \mathbf{L}^T\mathbf{L} \neq \mathbf{I}_M, \quad M > N \quad (2.32)$$

where \mathbf{I}_N and \mathbf{I}_M are the $N \times N$ and $M \times M$ identity matrices, respectively.

A matrix cannot be both orthogonal and semiorthogonal. Orthogonal matrices must be square, and semiorthogonal matrices cannot be square. Furthermore, if \mathbf{L} is a square $N \times N$ matrix, and

$$\mathbf{L}^T \mathbf{L} = \mathbf{I}_N \quad (2.26)$$

then it is not possible to have

$$\mathbf{L} \mathbf{L}^T \neq \mathbf{I}_N \quad (2.33)$$

2.2.3 Matrices and Vector Spaces

The columns or rows of a matrix can be thought of as vectors. For example, if \mathbf{A} is an $N \times M$ matrix, each column can be thought of as a vector in N -space because it has N entries. Conversely, each row of \mathbf{A} can be thought of as being a vector in M -space because it has M entries.

We note that for the linear system of equations given by

$$\mathbf{G} \mathbf{m} = \mathbf{d} \quad (1.13)$$

where \mathbf{G} is $N \times M$, \mathbf{m} is $M \times 1$, and \mathbf{d} is $N \times 1$, that the model parameter vector \mathbf{m} lies in M -space (along with all the rows of \mathbf{G}), while the data vector lies in N -space (along with all the columns of \mathbf{G}). In general, we will think of the $M \times 1$ vectors as lying in *model space*, while the $N \times 1$ vectors lie in *data space*.

Spanning a Space

The notion of spanning a space is important for any discussion of the uniqueness of solutions or of the ability to fit the data. We first need to introduce definitions of linear independence and vector orthogonality.

A set of M vectors \mathbf{v}_i , $i = 1, \dots, M$, in M -space (the set of all M -dimensional vectors), is said to be linearly independent if and only if

$$a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + \dots + a_M \mathbf{v}_M = \mathbf{0} \quad (2.34)$$

where a_i are constants, has only the trivial solution $a_i = 0$, $i = 1, \dots, M$.

This is equivalent to saying that an arbitrary vector \mathbf{s} in M space can be written as a linear combination of the \mathbf{v}_i , $i = 1, \dots, M$. That is, one can find a_i such that for an arbitrary vector \mathbf{s}

$$\mathbf{s} = a_1 \mathbf{v}_1 + a_2 \mathbf{v}_2 + \dots + a_M \mathbf{v}_M \quad (2.35)$$

Two vectors \mathbf{r} and \mathbf{s} in M -space are said to be orthogonal to each other if their dot, or inner, product with each other is zero. That is, if

$$\mathbf{r} \cdot \mathbf{s} = \|\mathbf{r}\| \|\mathbf{s}\| \cos \theta = 0 \quad (2.36)$$

where θ is the angle between the vectors, and $\|\mathbf{r}\|$, $\|\mathbf{s}\|$ are the lengths of \mathbf{r} and \mathbf{s} , respectively.

The dot product of two vectors is also given by

$$\mathbf{r}^T \mathbf{s} = \mathbf{s}^T \mathbf{r} = \sum_{i=1}^M r_i s_i \quad (2.37)$$

M space is spanned by any set of M linearly independent M -dimensional vectors.

Rank of a Matrix

The number of linearly independent rows in a matrix, which is also equal to the number of linearly independent columns, is called the rank of the matrix. The rank of matrices is defined for both square and nonsquare matrices. The rank of a matrix cannot exceed the minimum of the number of rows or columns in the matrix (i.e., the rank is less than or equal to the minimum of N, M).

If an $M \times M$ matrix is an orthogonal matrix, then it has rank M . The M rows are all linearly independent, as are the M columns. In fact, not only are the rows independent for an orthogonal matrix, they are orthogonal to each other. The same is true for the columns. If a matrix is semiorthogonal, then the M columns (or N rows, if $N < M$) are orthogonal to each other.

We will make extensive use of matrices and linear algebra in this course, especially when we work with the generalized inverse. Next, we need to turn our attention to probability and statistics.

2.3 Probability and Statistics

2.3.1 Introduction

We need some background in *probability* and *statistics* before proceeding very far. In this review section, I will cover the material from Menke's book, using some material from other math texts to help clarify things.

Basically, what we need is a way of describing the *noise* in data and estimated model parameters. We will need the following terms: *random variable*, *probability distribution*, *mean* or *expected value*, *maximum likelihood*, *variance*, *standard deviation*, *standardized normal variables*, *covariance*, *correlation coefficients*, *Gaussian distributions*, and *confidence intervals*.

2.3.2 Definitions, Part 1

Random Variable: A function that assigns a value to the outcome of an experiment. A random variable has well-defined properties based on some distribution. It is called random because you cannot know beforehand the exact value for the outcome of the experiment. One cannot measure directly the true properties of a random variable. One can only make measurements, also called *realizations*, of a random variable, and estimate its properties. The birth weight of baby goslings is a random variable, for example.

Probability Density Function: The true properties of a random variable b are specified by the *probability density function* $P(b)$. The probability that a particular realization of b will fall between b and $b + db$ is given by $P(b)db$. (Note that Menke uses d where I use b . His notation is bad when one needs to use integrals.) $P(b)$ satisfies

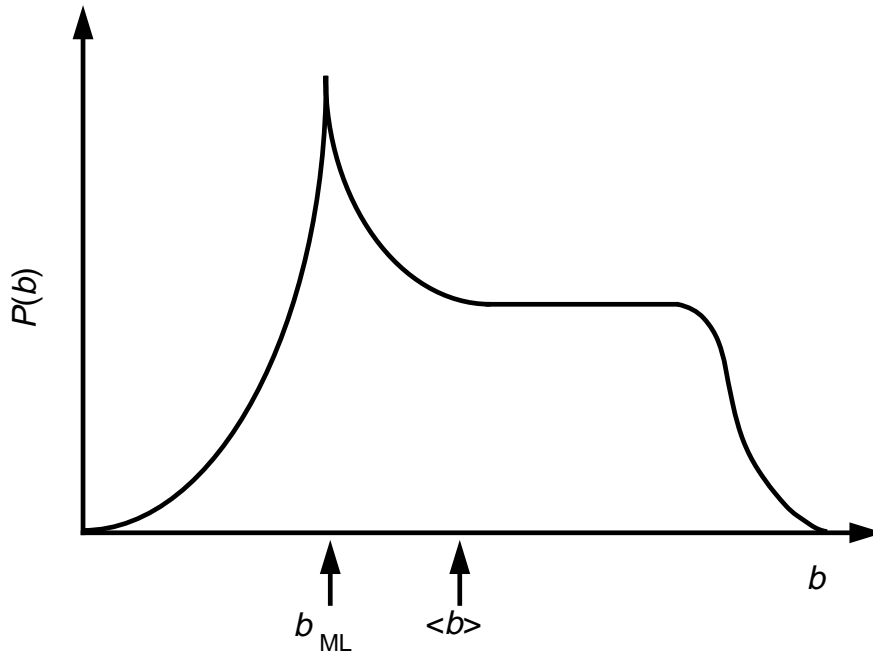
$$1 = \int_{-\infty}^{+\infty} P(b) db \quad (2.38)$$

which says that the probability of b taking on some value is 1. $P(b)$ completely describes the random variable b . It is often useful to try and find a way to summarize the properties of $P(b)$ with a few numbers, however.

Mean or Expected Value: The *mean value* $E(b)$ (also denoted $\langle b \rangle$) is much like the mean of a set of numbers; that is, it is the “balancing point” of the distribution $P(b)$ and is given by

$$E(b) = \int_{-\infty}^{+\infty} b P(b) db \quad (2.39)$$

Maximum Likelihood: This is the point in the probability distribution $P(b)$ that has the highest likelihood or probability. It may or may not be close to the mean $E(b) = \langle b \rangle$. An important point is that for Gaussian distributions, the maximum likelihood point and the mean $E(b) = \langle b \rangle$ are the same! The graph below (after Figure 2.3, p. 23, Menke) illustrates a case where the two are different.



The maximum likelihood point b_{ML} of the probability distribution $P(b)$ for data b gives the most probable value of the data. In general, this value can be different from the mean datum $\langle b \rangle$, which is at the “balancing point” of the distribution.

Variance: Variance is one measure of the spread, or width, of $P(b)$ about the mean $E(b)$. It is given by

$$\sigma^2 = \int_{-\infty}^{+\infty} (b - \langle b \rangle)^2 P(b) db \quad (2.40a)$$

Computationally, for L experiments in which the k th experiment gives b_k , the variance is given by

$$\sigma^2 = \frac{1}{L-1} \sum_{k=1}^L (b_k - \langle b \rangle)^2 \quad (2.40b)$$

Standard Deviation: Standard deviation is the positive square root of the variance, given by

$$\sigma = +\sqrt{\sigma^2} \quad (2.40c)$$

Covariance: Covariance is a measure of the correlation between errors. If the errors in two observations are uncorrelated, then the covariance is zero. We need another definition before proceeding.

Joint Density Function $P(\mathbf{b})$: The probability that b_1 is between b_1 and $b_1 + db_1$, that b_2 is between b_2 and $b_2 + db_2$, etc. If the data are independent, then

$$P(\mathbf{b}) = P(b_1) P(b_2) \cdots P(b_n) \quad (2.41)$$

If the data are correlated, then $P(\mathbf{b})$ will have some more complicated form. Then, the covariance between b_1 and b_2 is defined as

$$\text{cov}(b_1, b_2) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} (b_1 - \langle b_1 \rangle)(b_2 - \langle b_2 \rangle) P(\mathbf{b}) db_1 db_2 \cdots db_n \quad (2.42a)$$

In the event that the data are independent, this reduces to

$$\begin{aligned} \text{cov}(b_1, b_2) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (b_1 - \langle b_1 \rangle)(b_2 - \langle b_2 \rangle) P(b_1) P(b_2) db_1 db_2 \\ &= 0 \end{aligned} \quad (2.42b)$$

The reason is that for any value of $(b_1 - \langle b_1 \rangle)$, $(b_2 - \langle b_2 \rangle)$ is as likely to be positive as negative, i.e., the sum will average to zero. The matrix $[\text{cov } \mathbf{b}]$ contains all of the covariances defined using Equation (2.42) in an $N \times N$ matrix. Note also that the covariance of b_i with itself is just the variance of b_i .

In practical terms, if one has an N -dimensional data vector \mathbf{b} that has been measured L times, then the ij th term in $[\text{cov } \mathbf{b}]$, denoted $[\text{cov } \mathbf{b}]_{ij}$, is defined as

$$[\text{cov } \mathbf{b}]_{ij} = \frac{1}{L-1} \sum_{k=1}^L (b_i^k - \bar{b}_i)(b_j^k - \bar{b}_j) \quad (2.42c)$$

where b_i^k is the value of the i th datum in \mathbf{b} on the k th measurement of the data vector, $\langle b_i \rangle$ is the mean or average value of b_i for all L measurements, and the $L - 1$ term results from sampling theory.

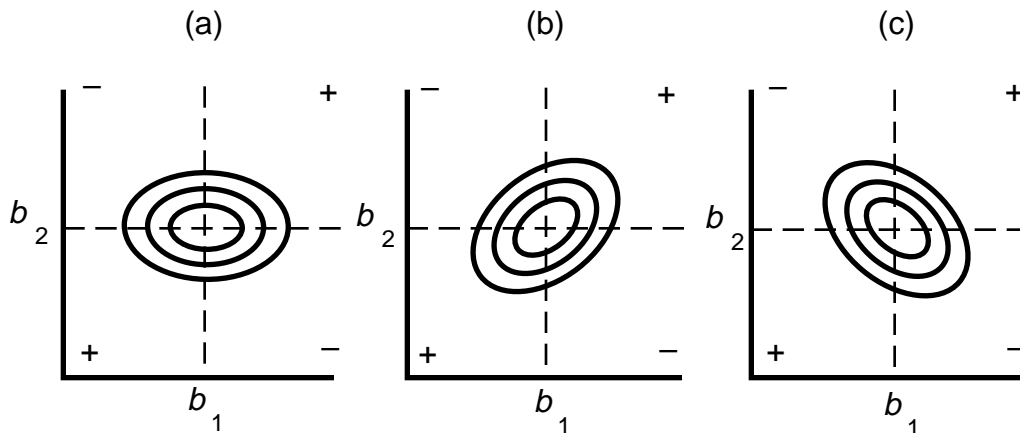
Correlation Coefficients: This is a normalized measure of the degree of correlation of errors. It takes on values between -1 and 1 , with a value of 0 implying no correlation.

The correlation coefficient matrix $[\text{cor } \mathbf{b}]$ is defined as

$$[\text{cor } \mathbf{b}]_{ij} = \frac{[\text{cov } \mathbf{b}]_{ij}}{\sigma_i \sigma_j} \quad (2.43)$$

where $[\text{cov } \mathbf{b}]_{ij}$ is the covariance matrix defined term by term as above for $\text{cov } [b_1, b_2]$, and σ_i, σ_j are the standard deviations for the i th and j th observations, respectively. The diagonal terms of $[\text{cor } \mathbf{b}]_{ij}$ are equal to 1 , since each observation is perfectly correlated with itself.

The figure below (after Figure 2.8, page 26, Menke) shows three different cases of degree of correlation for two observations b_1 and b_2 .



Contour plots of $P(b_1, b_2)$ when the data are (a) uncorrelated, (b) positively correlated, (c) negatively correlated. The dashed lines indicate the four quadrants of alternating sign used to determine correlation.

2.3.3 Some Comments on Applications to Inverse Theory

Some comments are now in order about the nature of the estimated model parameters. We will always assume that the noise in the observations can be described as random variables. Whatever inverse we create will map errors in the data into errors in the estimated model parameters. Thus, the estimated model parameters are themselves random variables. This is true even though the true model parameters may not be random variables. If the distribution of noise for the data is known, then in principle the distribution for the estimated model parameters can be found by “mapping” through the inverse operator.

This is often very difficult, but one particular case turns out to have a rather simple form. We will see where this form comes from when we get to the subject of generalized inverses. For now, consider the following as magic.

If the transformation between data \mathbf{b} and model parameters \mathbf{m} is of the form

$$\mathbf{m} = \mathbf{M}\mathbf{b} + \mathbf{v} \quad (2.44a)$$

where \mathbf{M} is any arbitrary matrix and \mathbf{v} is any arbitrary vector, then

$$\langle \mathbf{m} \rangle = \mathbf{M} \langle \mathbf{b} \rangle + \mathbf{v} \quad (2.44b)$$

and

$$[\text{cov } \mathbf{m}] = \mathbf{M} [\text{cov } \mathbf{b}] \mathbf{M}^T \quad (2.44c)$$

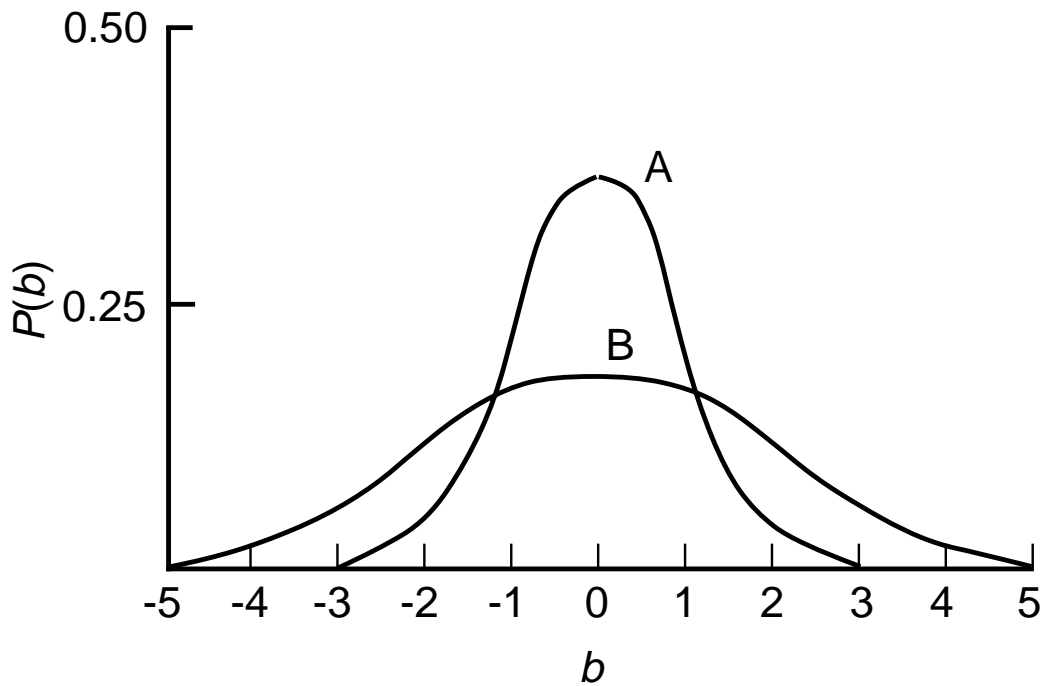
2.3.4 Definitions, Part 2

Gaussian Distribution: This is a particular probability distribution given by

$$P(b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(b - \langle b \rangle)^2}{2\sigma^2}\right] \quad (2.45a)$$

The figure below (after Figure 2.10, page 29, Menke) shows the familiar bell-shaped curve. It has the following properties:

$$\text{Mean} = E(b) = \langle b \rangle \quad \text{and} \quad \text{Variance} = \sigma^2$$



Gaussian distribution with zero mean and $\sigma = 1$ for curve A, and $\sigma = 2$ for curve B.

Many distributions can be approximated fairly accurately (especially away from the tails) by the Gaussian distribution. It is also very important because it is the limiting distribution for the sum of random variables. This is often just what one assumes for noise in the data.

One also needs a way to represent the joint probability introduced earlier for a set of random variables each of which has a Gaussian distribution. The joint probability density function

for a vector \mathbf{b} of observations that all have Gaussian distributions is chosen to be [see Equation (2.10) of Menke, page 30]

$$P(\mathbf{b}) = \frac{(\det[\text{cov } \mathbf{b}])^{-1/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{1}{2}[\mathbf{b} - \langle \mathbf{b} \rangle]^T [\text{cov } \mathbf{b}]^{-1} [\mathbf{b} - \langle \mathbf{b} \rangle]\right\} \quad (2.45b)$$

which reduces to the previous case in Equation (2.45a) for $N = 1$ and $\text{var}(b_1) = \sigma^2$. In statistics books, Equation (2.45b) is often given as

$$P(\mathbf{b}) = (2\pi)^{-N/2} |\Sigma_{\mathbf{b}}|^{-1/2} \exp\{-2[\mathbf{b} - \boldsymbol{\mu}_{\mathbf{b}}]^T \Sigma^{-1} [\mathbf{b} - \boldsymbol{\mu}_{\mathbf{b}}]\}$$

With this background, it makes sense (statistically, at least) to replace the original relationship:

$$\mathbf{b} = \mathbf{G}\mathbf{m} \quad (1.13)$$

with

$$\langle \mathbf{b} \rangle = \mathbf{G}\mathbf{m} \quad (2.46)$$

The reason is that one cannot expect that there is an \mathbf{m} that should exactly predict any particular realization of \mathbf{b} when \mathbf{b} is in fact a random variable.

Then the joint probability is given by

$$P(\mathbf{b}) = \frac{(\det[\text{cov } \mathbf{b}])^{-1/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{1}{2}[\mathbf{b} - \mathbf{G}\mathbf{m}]^T [\text{cov } \mathbf{b}]^{-1} [\mathbf{b} - \mathbf{G}\mathbf{m}]\right\} \quad (2.47)$$

What one then does is seek an \mathbf{m} that maximizes the probability that the predicted data are in fact close to the observed data. This is the basis of the *maximum likelihood* or probabilistic approach to inverse theory.

Standardized Normal Variables: It is possible to standardize random variables by subtracting their mean and dividing by the standard deviation.

If the random variable had a Gaussian (i.e., normal) distribution, then so does the standardized random variable. Now, however, the standardized normal variables have zero mean and standard deviation equal to one. Random variables can be standardized by the following transformation:

$$s = \frac{\mathbf{m} - \langle \mathbf{m} \rangle}{\sigma} \quad (2.48)$$

where you will often see \mathbf{z} replacing \mathbf{s} in statistics books.

We will see, when all is said and done, that most inverses represent a transformation to standardized variables, followed by a “simple” inverse analysis, and then a transformation back for the final solution.

Chi-Squared (Goodness of Fit) Test: A statistical test to see whether a particular observed distribution is likely to have been drawn from a population having some known form.

The application we will make of the chi-squared test is to test whether the noise in a particular problem is likely to have a Gaussian distribution. This is not the kind of question one can answer with certainty, so one must talk in terms of probability or likelihood. For example, in the chi-squared test, one typically says things like there is only a 5% chance that this sample distribution does not follow a Gaussian distribution.

As applied to testing whether a given distribution is likely to have come from a Gaussian population, the procedure is as follows: One sets up an arbitrary number of bins and compares the number of observations that fall into each bin with the number expected from a Gaussian distribution having the same mean and variance as the observed data. One quantifies the departure between the two distributions, called the chi-squared value and denoted χ^2 , as

$$\chi^2 = \sum_{i=1}^k \frac{[(\# \text{ obs in bin } i) - (\# \text{ expected in bin } i)]^2}{[\# \text{ expected in bin } i]} \quad (2.49)$$

where the sum is over the number of bins, k . Next, the number of degrees of freedom for the problem must be considered. For this problem, the number of degrees is equal to the number of bins minus three. The reason you subtract three is as follows: You subtract 1 because if an observation does not fall into any subset of $k - 1$ bins, you know it falls in the one bin left over. You are not free to put it anywhere else. The other two come from the fact that you have assumed that the mean and standard deviation of the observed data set are the mean and standard deviations for the theoretical Gaussian distribution.

With this information in hand, one uses standard chi-squared test tables from statistics books and determines whether such a departure would occur randomly more often than, say, 5% of the time. Officially, the null hypothesis is that the sample was drawn from a Gaussian distribution. If the observed value for χ^2 is greater than χ^2_{α} , then one rejects the null hypothesis at the α significance level. Typically, 0.05 is used for the test. The α significance level is equivalent to the $(1 - \alpha)\%$ confidence level.

Confidence Intervals: One says, for example, with 98% confidence that the true mean of a random variable lies between two values. This is based on knowing the probability distribution for the random variable, of course, and can be very difficult, especially for complicated distributions that include nonzero correlation coefficients. However, for Gaussian distributions, these are well known and can be found in any standard statistics book. For example, Gaussian distributions have 68% and 95% confidence intervals of approximately $\pm 1\sigma$ and $\pm 2\sigma$, respectively.

T and F Tests: These two statistical tests are commonly used to determine whether the properties of two samples are consistent with the samples coming from the same population.

The F test in particular can be used to test the improvement in the fit between predicted and observed data when one adds a degree of freedom in the inversion. One expects to fit the data better by adding more model parameters, so the relevant question is whether the improvement is significant.

As applied to the test of improvement in fit between case 1 and case 2, where case 2 uses more model parameters (degrees of freedom) to describe the same data set, the F ratio is given by

$$F = \frac{(E_1 - E_2)/(DOF_1 - DOF_2)}{(E_2 / DOF_2)} \quad (2.50)$$

where E is the residual sum of squares and DOF is the number of degrees of freedom for each case.

If F is large, one accepts that the second case with more model parameters provides a significantly better fit to the data. The calculated F is compared to published tables with $DOF_1 - DOF_2$ and DOF_2 degrees of freedom at a specified confidence level. (Reference: T. M. Hearn, P_n travel times in Southern California, *J. Geophys. Res.*, 89, 1843–1855, 1984.)

The next section will deal with solving inverse problems based on length measures. This will include the classic least squares approach.

CHAPTER 3: INVERSE METHODS BASED ON LENGTH

3.1 Introduction

This chapter is concerned with inverse methods based on the length of various vectors that arise in a typical problem. The two most common vectors concerned are the data error, or misfit, vector and the model parameter vector. Methods based on the first vector give rise to classic least squares solutions. Methods based on the second vector give rise to what are known as minimum length solutions. Improvements over simple least squares and minimum length solutions include the use of information about noise in the data and *a priori* information about the model parameters, and are known as weighted least squares or weighted minimum length solutions, respectively. This chapter will end with material on how to handle constraints and on variances of the estimated model parameters.

3.2 Data Error and Model Parameter Vectors

The data error and model parameter vectors will play an essential role in the development of inverse methods. They are given by

$$\text{data error vector} = \mathbf{e} = \mathbf{d}^{\text{obs}} - \mathbf{d}^{\text{pre}} \quad (3.1)$$

and

$$\text{model parameter vector} = \mathbf{m} \quad (3.2)$$

The dimension of the error vector \mathbf{e} is $N \times 1$, while the dimension of the model parameter vector is $M \times 1$, respectively. In order to utilize these vectors, we next consider the notion of the size, or length, of vectors.

3.3 Measures of Length

The *norm* of a vector is a measure of its size, or length. There are many possible definitions for norms. We are most familiar with the Cartesian (L_2) norm. Some examples of norms follow:

$$L_1 = \sum_{i=1}^N |e_i| \quad (3.3a)$$

$$L_2 = \left[\sum_{i=1}^N |e_i|^2 \right]^{1/2} \quad (3.3b)$$

$$L_M = \left[\sum_{i=1}^N |e_i|^M \right]^{1/M} \quad (3.3c)$$

and finally,

$$L_\infty = \max_i |e_i| \quad (3.3d)$$

Important Notice!

Inverse methods based on different norms can, and often do, give different answers!

The reason is that different norms give different *weight* to “outliers.” For example, the L_∞ norm gives all the weight to the largest misfit. Low-order norms give more equal weight to errors of different sizes.

The L_2 norm gives the familiar Cartesian length of a vector. Consider the total misfit E between observed and predicted data. It has units of length squared and can be found either as the square of the L_2 norm of \mathbf{e} , the error vector (Equation 3.1), or by noting that it is also equivalent to the dot (or inner) product of \mathbf{e} with itself, given by

$$E = \mathbf{e}^T \mathbf{e} = [e_1 \quad e_2 \quad \cdots \quad e_N] \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \sum_{i=1}^N e_i^2 \quad (3.4)$$

Inverse methods based on the L_2 norm are also closely tied to the notion that errors in the data have Gaussian statistics. They give considerable weight to large errors, which would be considered “unlikely” if, in fact, the errors were distributed in a Gaussian fashion.

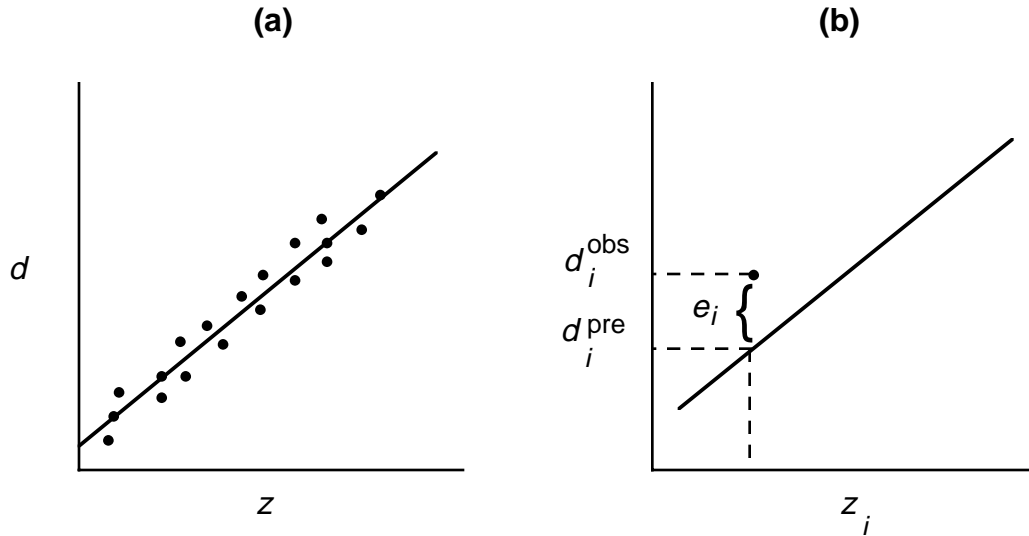
Now that we have a way to quantify the misfit between predicted and observed data, we are ready to define a procedure for estimating the value of the elements in \mathbf{m} . The procedure is to take the partial derivative of E with respect to each element in \mathbf{m} and set the resulting equations to zero. This will produce a system of M equations that can be manipulated in such a way that, in general, leads to a solution for the M elements of \mathbf{m} .

The next section will show how this is done for the least squares problem of finding a best fit straight line to a set of data points.

3.4 Minimizing the Misfit—Least Squares

3.4.1 Least Squares Problem for a Straight Line

Consider the figure below (after Figure 3.1 from Menke, page 36):



(a) Least squares fitting of a straight line to (z, d) pairs. (b) The error e_i for each observation is the difference between the observed and predicted datum: $e_i = d_i^{\text{obs}} - d_i^{\text{pre}}$.

The i th predicted datum d_i^{pre} for the straight line problem is given by

$$d_i^{\text{pre}} = m_1 + m_2 z_i \quad (3.5)$$

where the two unknowns, m_1 and m_2 , are the intercept and slope of the line, respectively, and z_i is the value along the z axis where the i th observation is made.

For N points we have a system of N such equations that can be written in matrix form as:

$$\begin{bmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} 1 & z_1 \\ \vdots & \vdots \\ 1 & z_i \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \quad (3.6)$$

Or, in the by now familiar matrix notation, as

$$\mathbf{d} = \mathbf{G} \mathbf{m} \quad (1.13)$$

$(N \times 1) \quad (N \times 2) \quad (2 \times 1)$

The total misfit E is given by

$$E = \mathbf{e}^T \mathbf{e} = \sum_i^N \left[d_i^{\text{obs}} - d_i^{\text{pre}} \right]^2 \quad (3.7)$$

$$= \sum_i^N \left[d_i^{\text{obs}} - (m_1 + m_2 z_i) \right]^2 \quad (3.8)$$

Dropping the “obs” in the notation for the observed data, we have

$$E = \sum_i^N \left[d_i^2 - 2d_i m_1 - 2d_i m_2 z_i + 2m_1 m_2 z_i + m_1^2 + m_2^2 z_i^2 \right] \quad (3.9)$$

Then, taking the partials of E with respect to m_1 and m_2 , respectively, and setting them to zero yields the following equations:

$$\frac{\partial E}{\partial m_1} = 2Nm_1 - 2 \sum_{i=1}^N d_i + 2m_2 \sum_{i=1}^N z_i = 0 \quad (3.10)$$

and

$$\frac{\partial E}{\partial m_2} = -2 \sum_{i=1}^N d_i z_i + 2m_1 \sum_{i=1}^N z_i + 2m_2 \sum_{i=1}^N z_i^2 = 0 \quad (3.11)$$

Rewriting Equations (3.10) and (3.11) above yields

$$Nm_1 + m_2 \sum_i z_i = \sum_i d_i \quad (3.12)$$

and

$$m_1 \sum_i z_i + m_2 \sum_i z_i^2 = \sum_i d_i z_i \quad (3.13)$$

Combining the two equations in matrix notation in the form $\mathbf{A}\mathbf{m} = \mathbf{b}$ yields

$$\begin{bmatrix} N & \sum z_i \\ \sum z_i & \sum z_i^2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} \sum d_i \\ \sum d_i z_i \end{bmatrix} \quad (3.14)$$

or simply

$$\begin{matrix} \mathbf{A} & \mathbf{m} & = & \mathbf{b} \\ (2 \times 2) & (2 \times 1) & & (2 \times 1) \end{matrix} \quad (3.15)$$

Note that by the above procedure we have reduced the problem from one with N equations in two unknowns (m_1 and m_2) in $\mathbf{Gm} = \mathbf{d}$ to one with two equations in the same two unknowns in $\mathbf{Am} = \mathbf{b}$.

The matrix equation $\mathbf{Am} = \mathbf{b}$ can also be rewritten in terms of the original \mathbf{G} and \mathbf{d} when one notices that the matrix \mathbf{A} can be factored as

$$\begin{matrix} \begin{bmatrix} N & \Sigma z_i \\ \Sigma z_i & \Sigma z_i^2 \end{bmatrix} & = & \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} & \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} & = & \mathbf{G}^T \mathbf{G} \\ (2 \times 2) & & (2 \times N) & & (N \times 2) & (2 \times 2) \end{matrix} \quad (3.16)$$

Also, \mathbf{b} above can be rewritten similarly as

$$\begin{bmatrix} \Sigma d_i \\ \Sigma d_i z_i \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \mathbf{G}^T \mathbf{d} \quad (3.17)$$

Thus, substituting Equations (3.16) and (3.17) into Equation (3.14), one arrives at the so-called *normal equations* for the least squares problem:

$$\mathbf{G}^T \mathbf{Gm} = \mathbf{G}^T \mathbf{d} \quad (3.18)$$

The least squares solution \mathbf{m}_{LS} is then found as

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.19)$$

assuming that $[\mathbf{G}^T \mathbf{G}]^{-1}$ exists.

In summary, we used the forward problem (Equation 3.6) to give us an explicit relationship between the model parameters (m_1 and m_2) and a measure of the misfit to the observed data, E . Then, we minimized E by taking the partial derivatives of the misfit function with respect to the unknown model parameters, setting the partials to zero, and solving for the model parameters.

3.4.2 Derivation of the General Least Squares Solution

We start with any system of linear equations which can be expressed in the form

$$\begin{matrix} \mathbf{d} & = & \mathbf{G} & \mathbf{m} \\ (N \times 1) & & (N \times M) & (M \times 1) \end{matrix} \quad (1.13)$$

Again, let $E = \mathbf{e}^T \mathbf{e} = [\mathbf{d} - \mathbf{d}^{pre}]^T [\mathbf{d} - \mathbf{d}^{pre}]$

$$E = [\mathbf{d} - \mathbf{Gm}]^T [\mathbf{d} - \mathbf{Gm}] \quad (3.20)$$

$$E = \sum_{i=1}^N \left[d_i - \sum_{j=1}^M G_{ij} m_j \right] \left[d_i - \sum_{k=1}^M G_{ik} m_k \right] \quad (3.21)$$

As before, the procedure is to write out the above equation with all its cross terms, take partials of E with respect to each of the elements in \mathbf{m} , and set the corresponding equations to zero. For example, following Menke, page 40, Equations (3.6)–(3.9), we obtain an expression for the partial of E with respect to m_q :

$$\frac{\partial E}{\partial m_q} = 2 \sum_{k=1}^M m_k \sum_{i=1}^N G_{iq} G_{ik} - 2 \sum_{i=1}^N G_{iq} d_i = 0 \quad (3.22a)$$

We can simplify this expression by recalling Equation (2.4) from the introductory remarks on matrix manipulations in Chapter 2:

$$C_{ij} = \sum_{k=1}^M a_{ik} b_{kj} \quad (2.4)$$

Note that the first summation on i in Equation (3.22a) looks similar in form to Equation (2.4), but the subscripts on the first \mathbf{G} term are “backwards.” If we further note that interchanging the subscripts is equivalent to taking the transpose of \mathbf{G} , we see that the summation on i gives the qk -th entry in $\mathbf{G}^T \mathbf{G}$:

$$\sum_{i=1}^N G_{iq} G_{ik} = \sum_{i=1}^N [G^T]_{qi} G_{ik} = [\mathbf{G}^T \mathbf{G}]_{qk} \quad (3.22b)$$

Thus, Equation (3.22a) reduces to

$$\frac{\partial E}{\partial m_q} = 2 \sum_{k=1}^M m_k [\mathbf{G}^T \mathbf{G}]_{qk} - 2 \sum_{i=1}^N G_{iq} d_i = 0 \quad (3.23)$$

Now, we can further simplify the first summation by recalling Equation (2.6) from the same section

$$d_i = \sum_{j=1}^M G_{ij} m_j \quad (2.6)$$

To see this clearly, we rearrange the order of terms in the first sum as follows:

$$\sum_{k=1}^M m_k [\mathbf{G}^T \mathbf{G}]_{qk} = \sum_{k=1}^M [\mathbf{G}^T \mathbf{G}]_{qk} m_k = [\mathbf{G}^T \mathbf{G} \mathbf{m}]_q \quad (3.24)$$

which is the q th entry in $\mathbf{G}^T \mathbf{G} \mathbf{m}$. Note that $\mathbf{G}^T \mathbf{G} \mathbf{m}$ has dimension $(M \times N)(N \times M)(M \times 1) = (M \times 1)$. That is, it is an M -dimensional vector.

In a similar fashion, the second summation on i can be reduced to a term in $[\mathbf{G}^T \mathbf{d}]_q$, the q th entry in an $(M \times N)(N \times 1) = (M \times 1)$ dimensional vector. Thus, for the q th equation, we have

$$0 = \frac{\partial E}{\partial m_q} = 2[\mathbf{G}^T \mathbf{G} \mathbf{m}]_q - 2[\mathbf{G}^T \mathbf{d}]_q \quad (3.25)$$

Dropping the common factor of 2 and combining the q equations into matrix notation, we arrive at

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (3.26)$$

The least squares solution for \mathbf{m} is thus given by

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

The least squares operator, \mathbf{G}_{LS}^{-1} , is thus given by

$$\mathbf{G}_{LS}^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.28)$$

Recalling basic calculus, we note that \mathbf{m}_{LS} above is the solution that minimizes E , the total misfit. Summarizing, setting the q partial derivatives of E with respect to the elements in \mathbf{m} to zero leads to the least squares solution.

We have just derived the least squares solution by taking the partial derivatives of E with respect to m_q and then combining the terms for $q = 1, 2, \dots, M$. An alternative, but equivalent, formulation begins with Equation (3.2) but is written out as

$$E = [\mathbf{d} - \mathbf{G} \mathbf{m}]^T [\mathbf{d} - \mathbf{G} \mathbf{m}] \quad (3.20)$$

$$= [\mathbf{d}^T - \mathbf{m}^T \mathbf{G}^T] [\mathbf{d} - \mathbf{G} \mathbf{m}]$$

$$= \mathbf{d}^T \mathbf{d} - \mathbf{d}^T \mathbf{G} \mathbf{m} - \mathbf{m}^T \mathbf{G}^T \mathbf{d} + \mathbf{m}^T \mathbf{G}^T \mathbf{G} \mathbf{m} \quad (3.29)$$

Then, taking the partial derivative of E with respect to \mathbf{m}^T turns out to be equivalent to what was done in Equations (3.22)–(3.26) for m_q , namely

$$\partial E / \partial \mathbf{m}^T = -\mathbf{G}^T \mathbf{d} + \mathbf{G}^T \mathbf{G} \mathbf{m} = 0 \quad (3.30)$$

which leads to

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (3.26)$$

and

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

It is also perhaps interesting to note that we could have obtained the same solution without taking partials. To see this, consider the following four steps.

Step 1. We begin with

$$\mathbf{G} \mathbf{m} = \mathbf{d} \quad (1.13)$$

Step 2. We then premultiply both sides by \mathbf{G}^T

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (3.26)$$

Step 3. Premultiply both sides by $[\mathbf{G}^T \mathbf{G}]^{-1}$

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{G} \mathbf{m} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.31)$$

Step 4. This reduces to

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

as before. The point is, however, that this way does not show why \mathbf{m}_{LS} is the solution which minimizes E , the misfit between the observed and predicted data.

All of this assumes that $[\mathbf{G}^T \mathbf{G}]^{-1}$ exists, of course. We will return to the existence and properties of $[\mathbf{G}^T \mathbf{G}]^{-1}$ later. Next, we will look at two examples of least squares problems to show a striking similarity that is not obvious at first glance.

3.4.3 Two Examples of Least Squares Problems

Example 1. Best-Fit Straight-Line Problem

We have, of course, already derived the solution for this problem in the last section. Briefly, then, for the system of equations

$$\mathbf{d} = \mathbf{G} \mathbf{m} \quad (1.13)$$

given by

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \quad (3.6)$$

we have

$$\mathbf{G}^T \mathbf{G} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_N \end{bmatrix} = \begin{bmatrix} N & \sum z_i \\ \sum z_i & \sum z_i^2 \end{bmatrix} \quad (3.32a)$$

and

$$\mathbf{G}^T \mathbf{d} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} \Sigma d_i \\ \Sigma d_i z_i \end{bmatrix} \quad (3.32b)$$

Thus, the least squares solution is given by

$$\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}_{\text{LS}} = \begin{bmatrix} N & \Sigma z_i \\ \Sigma z_i & \Sigma z_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \Sigma d_i \\ \Sigma d_i z_i \end{bmatrix} \quad (3.32c)$$

Example 2. Best-Fit Parabola Problem

The i th predicted datum for a parabola is given by

$$d_i = m_1 + m_2 z_i + m_3 z_i^2 \quad (3.33)$$

where m_1 and m_2 have the same meanings as in the straight line problem, and m_3 is the coefficient of the quadratic term. Again, the problem can be written in the form:

$$\mathbf{d} = \mathbf{Gm} \quad (1.13)$$

where now we have

$$\begin{bmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} 1 & z_1 & z_1^2 \\ \vdots & \vdots & \vdots \\ 1 & z_i & z_i^2 \\ \vdots & \vdots & \vdots \\ 1 & z_N & z_N^2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (3.34)$$

and

$$\mathbf{G}^T \mathbf{G} = \begin{bmatrix} N & \Sigma z_i & \Sigma z_i^2 \\ \Sigma z_i & \Sigma z_i^2 & \Sigma z_i^3 \\ \Sigma z_i^2 & \Sigma z_i^3 & \Sigma z_i^4 \end{bmatrix}, \quad \mathbf{G}^T \mathbf{d} = \begin{bmatrix} \Sigma d_i \\ \Sigma d_i z_i \\ \Sigma d_i z_i^2 \end{bmatrix} \quad (3.35)$$

As before, we form the least squares solution as

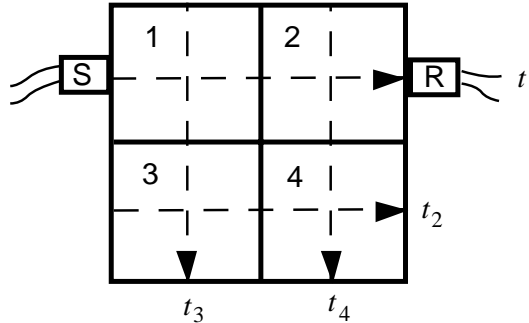
$$\mathbf{m}_{\text{LS}} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

Although the forward problems of predicting data for the straight line and parabolic cases look very different, the least squares solution is formed in a way that emphasizes the fundamental similarity between the two problems. For example, notice how the straight line problem is buried within the parabola problem. The upper left hand 2×2 part of $\mathbf{G}^T \mathbf{G}$ in Equation (3.35) is the same as Equation (3.32a). Also, the first two entries in $\mathbf{G}^T \mathbf{d}$ in Equation (3.35) are the same as Equation (3.32b).

Next we consider a four-parameter example.

3.4.4 Four-Parameter Tomography Problem

Finally, let's consider a four-parameter problem, but this one based on the concept of tomography.



$$\begin{aligned}
 t_1 &= h\left(\frac{1}{v_1}\right) + h\left(\frac{1}{v_2}\right) = h(s_1 + s_2) \\
 t_2 &= h\left(\frac{1}{v_3}\right) + h\left(\frac{1}{v_4}\right) = h(s_3 + s_4) \\
 t_3 &= h\left(\frac{1}{v_1}\right) + h\left(\frac{1}{v_3}\right) = h(s_1 + s_3) \\
 t_4 &= h\left(\frac{1}{v_2}\right) + h\left(\frac{1}{v_4}\right) = h(s_2 + s_4)
 \end{aligned} \tag{3.36}$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = h \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{3.37}$$

or

$$\mathbf{d} = \mathbf{Gm} \tag{1.13}$$

$$\mathbf{G}^T \mathbf{G} = h^2 \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = h^2 \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \tag{3.38}$$

$$\mathbf{G}^T \mathbf{d} = h \begin{bmatrix} t_1 + t_3 \\ t_1 + t_4 \\ t_2 + t_3 \\ t_2 + t_4 \end{bmatrix} \tag{3.39}$$

So, the normal equations are

$$\mathbf{G}^T \mathbf{Gm} = \mathbf{G}^T \mathbf{d} \tag{3.18}$$

$$h \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} t_1 + t_3 \\ t_1 + t_4 \\ t_2 + t_3 \\ t_2 + t_4 \end{bmatrix} \quad (3.40)$$

or

$$h \left(\begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} s_{1+} \quad \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} s_{2+} \quad \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix} s_{3+} \quad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 2 \end{bmatrix} s_4 \right) = \begin{bmatrix} t_1 + t_3 \\ t_1 + t_4 \\ t_2 + t_3 \\ t_2 + t_4 \end{bmatrix} \quad (3.41)$$

Example: $s_1 = s_2 = s_3 = s_4 = 1$, $h = 1$; then $t_1 = t_2 = t_3 = t_4 = 1$

By inspection, $s_1 = s_2 = s_3 = s_4 = 1$ is a solution, *but* so is $s_1 = s_4 = 2$, $s_2 = s_3 = 0$, or $s_1 = s_4 = 0$, $s_2 = s_3 = 2$.

Solutions are nonunique! Look back at \mathbf{G} . Are all of the columns or rows independent? No! What does that imply about \mathbf{G} (and $\mathbf{G}^T\mathbf{G}$)? Rank < 4. What does that imply about $(\mathbf{G}^T\mathbf{G})^{-1}$? It does not exist. So does \mathbf{m}_{LS} exist? No.

Other ways of saying this: The vectors \mathbf{g}_i do not span the space of \mathbf{m} . Or, the experimental set-up is not sufficient to uniquely determine the solution. Note that this analysis can be done without any data, based strictly on the experimental design.

Another way to look at it: Are the columns of \mathbf{G} independent? No. For example, coefficients -1 , $+1$, $+1$, and -1 will make the equations add to zero. What pattern does that suggest is not resolvable?

Now that we have derived the least squares solution, and considered some examples, we next turn our attention to something called the determinacy of the system of equations given by Equation (1.13):

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

This will begin to permit us to classify systems of equations based on the nature of \mathbf{G} .

3.5 Determinacy of Least Squares Problems

(See Pages 46–52, Menke)

3.5.1 Introduction

We have seen that the least squares solution to $\mathbf{d} = \mathbf{G}\mathbf{m}$ is given by

$$\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d} \quad (3.27)$$

There is no guarantee, as we saw in Section 3.4.4, that the solution even exists. It fails to exist when the matrix $\mathbf{G}^T\mathbf{G}$ has no mathematical inverse. We note that $\mathbf{G}^T\mathbf{G}$ is square ($M \times M$), and it is at least mathematically possible to consider inverting $\mathbf{G}^T\mathbf{G}$. (N.B. The dimension of $\mathbf{G}^T\mathbf{G}$ is $M \times M$, independent of the number of observations made). Mathematically, we can say the $\mathbf{G}^T\mathbf{G}$ has an inverse, and it is unique, when $\mathbf{G}^T\mathbf{G}$ has rank M . The rank of a matrix was considered in Section 2.2.3. Essentially, if $\mathbf{G}^T\mathbf{G}$ has rank M , then it has enough information in it to “resolve” M things (in this case, model parameters). This happens when all M rows (or equivalently, since $\mathbf{G}^T\mathbf{G}$ is square, all M columns) are independent. Recall also that independent means you cannot write any row (or column) as a linear combination of the other rows (columns).

$\mathbf{G}^T\mathbf{G}$ will have rank $< M$ if the number of observations N is less than M . Menke gives the example (pp. 45–46) of the straight-line fit to a single data point as an illustration. If $[\mathbf{G}^T\mathbf{G}]^{-1}$ does not exist, an infinite number of estimates will all fit the data equally well. Mathematically, $\mathbf{G}^T\mathbf{G}$ has rank $< M$ if $|\mathbf{G}^T\mathbf{G}| = 0$, where $|\mathbf{G}^T\mathbf{G}|$ is the determinant of $\mathbf{G}^T\mathbf{G}$.

Now, let us introduce Menke’s nomenclature based on the nature of $\mathbf{G}^T\mathbf{G}$ and on the prediction error. In all cases, the number of model parameters is M and the number of observations is N .

3.5.2 Even-Determined Problems: $M = N$

If a solution exists, it is unique. The prediction error $[\mathbf{d}^{\text{obs}} - \mathbf{d}^{\text{pre}}]$ is identically zero. For example,

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 5 & -1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \quad (3.42)$$

for which the solution is $\mathbf{m} = [1, 3]^T$.

3.5.3 Overdetermined Problems: Typically, $N > M$

With more observations than unknowns, typically one cannot fit all the data exactly. The least squares problem falls in this category. Consider the following example:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 5 & -1 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \quad (3.43)$$

This overdetermined case consists of adding one equation to Equation (3.42) in the previous example. The least squares solution is $[1.333, 4.833]^T$. The data can no longer be fit exactly.

3.5.4 Underdetermined Problems: Typically, $M > N$

With more unknowns than observations, \mathbf{m} has no unique solution. A special case of the underdetermined problem occurs when you can fit the data exactly, which is called the *purely*

underdetermined case. The prediction error for the purely underdetermined case is exactly zero (i.e., the data can be fit exactly). An example of such a problem is

$$[1] = [2 \quad 1] \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \quad (3.44)$$

Possible solutions include $[0, 1]^T$, $[0.5, 0]^T$, $[5, -9]^T$, $[1/3, 1/3]^T$ and $[0.4, 0.2]^T$. The solution with the minimum length, in the L_2 norm sense, is $[0.4, 0.2]^T$.

The following example, however, is also underdetermined, but no choice of m_1, m_2, m_3 will produce zero prediction error. Thus, it is not purely underdetermined.

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (3.45)$$

(You might want to verify the above examples. Can you think of others?)

Although I have stated that overdetermined (underdetermined) problems typically have $N > M$ ($N < M$), it is important to realize that this is not always the case. Consider the following:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (3.46)$$

For this problem, m_1 is *overdetermined*, (that is, no choice of m_1 can exactly fit both d_1 and d_2 unless d_1 happens to equal d_2), while at the same time m_2 and m_3 are *underdetermined*. This is the case even though there are two equations (i.e., the last two) in only two unknowns (m_2, m_3). The two equations, however, are not independent, since two times the next to last row in \mathbf{G} equals the last row. Thus this problem is both *overdetermined* and *underdetermined* at the same time.

For this reason, I am not very satisfied with Menke's nomenclature. As we will see later, when we deal with vector spaces, the key will be the single values (much like eigenvalues) and associated eigenvectors for the matrix \mathbf{G} .

3.6 Minimum Length Solution

The minimum length solution arises from the purely underdetermined case. In this section, we will develop the minimum length operator, using Lagrange multipliers and borrowing on the basic ideas of minimizing the length of a vector introduced in Section 3.4 on least squares.

3.6.1 Background Information

We begin with two pieces of information:

1. First, $[\mathbf{G}^T\mathbf{G}]^{-1}$ does not exist. Therefore, we cannot calculate the least squares solution $\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d}$.
2. Second, the prediction error $\mathbf{e} = \mathbf{d}^{obs} - \mathbf{d}^{pre}$ is exactly equal to zero.

To solve *underdetermined* problems, we must *add* information that is not already in \mathbf{G} . This is called *a priori* information. Examples might include the constraint that density be greater than zero for rocks, or that v_n , the seismic *P*-wave velocity at the Moho falls within the range $5 < v_n < 10$ km/s, etc.

Another *a priori* assumption is called “solution simplicity.” One seeks solutions that are as “simple” as possible. By analogy to seeking a solution with the “simplest” misfit to the data (i.e., the smallest) in the least squares problem, one can seek a solution which minimizes the total length of the model parameter vector, \mathbf{m} . At first glance, there may not seem to be any reason to do this. It does make sense for some cases, however. Suppose, for example, that the unknown model parameters are the velocities of points in a fluid. A solution that minimized the length of \mathbf{m} would also minimize the kinetic energy of the system. Thus, it would be appropriate in this case to minimize \mathbf{m} . It also turns out to be a nice property when one is doing nonlinear problems, and the \mathbf{m} that one is using is actually a vector of changes to the solution at the previous step. Then it is nice to have small step sizes. The requirement of solution simplicity will lead us, as shown later, to the so-called minimum length solution.

3.6.2 Lagrange Multipliers (See Page 50 and Appendix A.1, Menke)

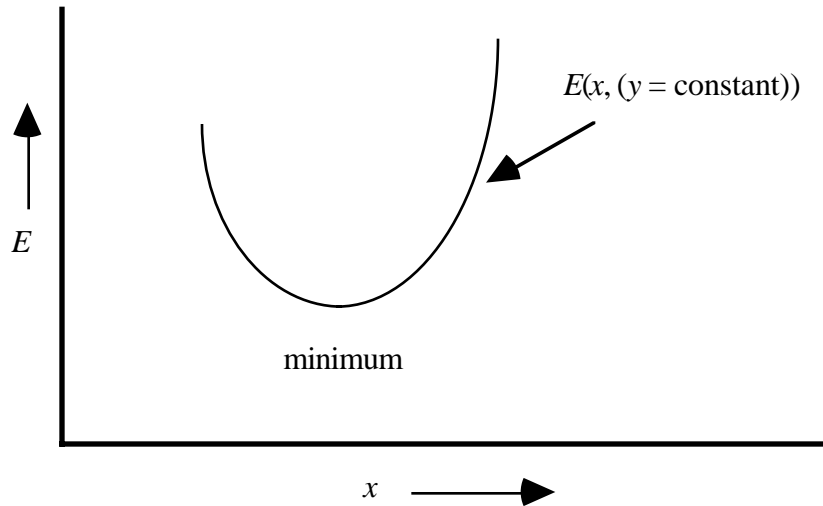
Lagrange multipliers come to mind whenever one wishes to solve a problem subject to some constraints. In the purely underdetermined case, these constraints are that the data misfit be zero. Before considering the full purely underdetermined case, consider the following discussion of Lagrange Multipliers, mostly after Menke.

Lagrange Multipliers With 2 Unknowns and 1 Constraint

Consider $E(x, y)$, a function of two variables. Suppose that we want to minimize $E(x, y)$ subject to some constraint of the form $\phi(x, y) = 0$.

The steps, using Lagrange multipliers, are as follows (next page).

Step 1. At the minimum in E , small changes in x and y lead to no change in E :



$$\therefore dE = \frac{\partial E}{\partial x} dx + \frac{\partial E}{\partial y} dy = 0 \quad (3.47)$$

Step 2. The constraint equation, however, says that dx and dy cannot be varied independently (since the constraint equation is independent, or different, from E). Since $\phi(x, y) = 0$ for all x, y , then so must $d\phi(x, y) = 0$. But,

$$d\phi = \frac{\partial \phi}{\partial x} dx + \frac{\partial \phi}{\partial y} dy = 0 \quad (3.48)$$

Step 3. Form the weighted sum of (3.47) and (3.48) as

$$dE + \lambda d\phi = \left(\frac{\partial E}{\partial x} + \lambda \frac{\partial \phi}{\partial x} \right) dx + \left(\frac{\partial E}{\partial y} + \lambda \frac{\partial \phi}{\partial y} \right) dy = 0 \quad (3.49)$$

where λ is a constant. Note that (3.49) holds for arbitrary λ .

Step 4. If λ is chosen, however, in such a way that

$$\frac{\partial E}{\partial x} + \lambda \frac{\partial \phi}{\partial x} = 0 \quad (3.50)$$

then it follows that

$$\frac{\partial E}{\partial y} + \lambda \frac{\partial \phi}{\partial y} = 0 \quad (3.51)$$

since at least one of dx, dy (in this case, dy) is arbitrary (i.e., dy may be chosen nonzero).

When λ has been chosen as indicated above, it is called the Lagrange multiplier. Therefore, (3.49) above is equivalent to minimizing $E + \lambda\phi$ without any constraints, i.e.,

$$\frac{\partial}{\partial x}(E + \lambda\phi) = \frac{\partial E}{\partial x} + \lambda \frac{\partial \phi}{\partial x} = 0 \quad (3.52)$$

and

$$\frac{\partial}{\partial y}(E + \lambda\phi) = \frac{\partial E}{\partial y} + \lambda \frac{\partial \phi}{\partial y} = 0 \quad (3.53)$$

Step 5. Finally, one must still solve the constraint equation

$$\phi(x, y) = 0 \quad (3.54)$$

Thus, the solution for (x, y) that minimizes E subject to the constraint that $\phi(x, y) = 0$ is given by (3.52), (3.53), and (3.54).

That is, the problem has reduced to the following three equations:

$$\frac{\partial E}{\partial x} + \lambda \frac{\partial \phi}{\partial x} = 0 \quad (3.50)$$

$$\frac{\partial E}{\partial y} + \lambda \frac{\partial \phi}{\partial y} = 0 \quad (3.51)$$

and

$$\phi(x, y) = 0 \quad (3.54)$$

in the three unknowns (x, y, λ) .

Extending the Problem to M Unknowns and N Constraints

The above procedure, used for a problem with two variables and one constraint, can be generalized to M unknowns in a vector \mathbf{m} subject to N constraints $\phi_j(\mathbf{m}) = 0, j = 1, \dots, N$. This leads to the following system of M equations, $i = 1, \dots, M$:

$$\frac{\partial E}{\partial m_i} + \sum_{j=1}^N \lambda_j \frac{\partial \phi_j}{\partial m_i} = 0 \quad (3.55)$$

with N constraints of the form

$$\phi_j(\mathbf{m}) = 0 \quad (3.56)$$

3.6.3 Application to the Purely Underdetermined Problem

With the background we now have in Lagrange multipliers, we are ready to reconsider the purely underdetermined problem. First, we pose the following problem: find \mathbf{m} such that $\mathbf{m}^T \mathbf{m}$ is minimized subject to the N constraints that the data misfit be zero.

$$e_i = d_i^{\text{obs}} - d_i^{\text{pre}} = d_i^{\text{obs}} - \sum_{j=1}^M G_{ij} m_j = 0 \quad (3.57)$$

That is, minimize

$$\psi(\mathbf{m}) = \mathbf{m}^T \mathbf{m} + \sum_{i=1}^N \lambda_i e_i \quad (3.58)$$

with respect to the elements m_i in \mathbf{m} . We can expand the terms in Equation (3.58) and obtain

$$\psi(\mathbf{m}) = \sum_{k=1}^M m_k^2 + \sum_{i=1}^N \lambda_i \left[d_i - \sum_{j=1}^M G_{ij} m_j \right] \quad (3.59)$$

Then, we have

$$\frac{\partial \psi}{\partial m_q} = 2 \sum_{k=1}^M \frac{\partial m_k}{\partial m_q} m_k - \sum_{i=1}^N \lambda_i \sum_{j=1}^M G_{ij} \frac{\partial m_j}{\partial m_q} \quad (3.60)$$

but

$$\frac{\partial m_k}{\partial m_q} = \delta_{kq} \quad \text{and} \quad \frac{\partial m_j}{\partial m_q} = \delta_{jq} \quad (3.61)$$

where δ_{ij} is the Kronecker delta, given by

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Thus

$$\frac{\partial \psi}{\partial m_q} = 2m_q - \sum_{i=1}^N \lambda_i G_{iq} = 0 \quad q = 1, 2, \dots, M \quad (3.62)$$

In matrix notation over all q , Equation (3.62) can be written as

$$2\mathbf{m} - \mathbf{G}^T \boldsymbol{\lambda} = \mathbf{0} \quad (3.63)$$

where $\boldsymbol{\lambda}$ is an $N \times 1$ vector containing the N Lagrange Multipliers $\lambda_i, i = 1, \dots, N$. Note that $\mathbf{G}^T \boldsymbol{\lambda}$ has dimension $(M \times N) \times (N \times 1) = M \times 1$, as required to be able to subtract it from \mathbf{m} .

Now, solving explicitly for \mathbf{m} yields

$$\mathbf{m} = \frac{1}{2} \mathbf{G}^T \boldsymbol{\lambda} \quad (3.64)$$

The constraints in this case are that the data be fit exactly. That is,

$$\mathbf{d} = \mathbf{Gm} \quad (1.13)$$

Substituting (3.64) into (1.13) gives

$$\mathbf{d} = \mathbf{Gm} = \mathbf{G}\left(\frac{1}{2} \mathbf{G}^T \boldsymbol{\lambda}\right) \quad (3.65)$$

which implies

$$\mathbf{d} = \frac{1}{2} \mathbf{GG}^T \boldsymbol{\lambda}$$

where \mathbf{GG}^T has dimension $(N \times M) \times (M \times N)$, or simply $N \times N$. Solving for $\boldsymbol{\lambda}$, when $[\mathbf{GG}^T]^{-1}$ exists, yields

$$\boldsymbol{\lambda} = 2[\mathbf{GG}^T]^{-1} \mathbf{d} \quad (3.66)$$

The Lagrange Multipliers are not ends in and of themselves. But, upon substitution of Equation (3.66) into (3.64), we obtain

$$\mathbf{m} = \frac{1}{2} \mathbf{G}^T \boldsymbol{\lambda} = \frac{1}{2} \mathbf{G}^T \{2[\mathbf{GG}^T]^{-1}\} \mathbf{d} \quad (3.6)$$

Rearranging, we arrive at the *minimum length solution*, \mathbf{m}_{ML} :

$$\mathbf{m}_{\text{ML}} = \mathbf{G}^T [\mathbf{GG}^T]^{-1} \mathbf{d} \quad (3.68)$$

where \mathbf{GG}^T is an $N \times N$ matrix and the minimum length operator, $\mathbf{G}_{\text{ML}}^{-1}$, is given by

$$\mathbf{G}_{\text{ML}}^{-1} = \mathbf{G}^T [\mathbf{GG}^T]^{-1} \quad (3.69)$$

The above procedure, then, is one that determines the solution which has the minimum length (L_2 norm = $[\mathbf{m}^T \mathbf{m}]^{1/2}$) amongst the infinite number of solutions that fit the data exactly. In practice, one does not actually calculate the values of the Lagrange multipliers, but goes directly to (3.68) above.

The above derivation shows that the length of \mathbf{m} is minimized by the minimum length operator. It may make more sense to seek a solution that deviates as little as possible from some prior estimate of the solution, $\langle \mathbf{m} \rangle$, rather than from zero. The zero vector is, in fact, the prior estimate $\langle \mathbf{m} \rangle$ for the minimum length solution given in Equation (3.68). If we wish to explicitly include $\langle \mathbf{m} \rangle$, then Equation (3.68) becomes

$$\begin{aligned} \mathbf{m}_{ML} &= \langle \mathbf{m} \rangle + \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} [\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] \\ &= \langle \mathbf{m} \rangle + \mathbf{G}_{ML}^{-1} [\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] = \mathbf{G}_{ML}^{-1} \mathbf{d} + [\mathbf{I} - \mathbf{G}_{ML}^{-1} \mathbf{G}] \langle \mathbf{m} \rangle \end{aligned} \quad (3.70)$$

We note immediately that Equation (3.70) reduces to Equation (3.68) when $\langle \mathbf{m} \rangle = \mathbf{0}$.

3.6.4 Comparison of Least Squares and Minimum Length Solutions

In closing this section, it is instructive to note the similarity in form between the minimum length and least squares solutions:

$$\text{Least Squares:} \quad \mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

$$\text{with} \quad \mathbf{G}_{LS}^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.28)$$

$$\text{Minimum Length:} \quad \mathbf{m}_{ML} = \langle \mathbf{m} \rangle + \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} [\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] \quad (3.70)$$

$$\text{with} \quad \mathbf{G}_{ML}^{-1} = \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.69)$$

The minimum length solution exists when $[\mathbf{G}\mathbf{G}^T]^{-1}$ exists. Since $\mathbf{G}\mathbf{G}^T$ is $N \times N$, this is the same as saying when $\mathbf{G}\mathbf{G}^T$ has rank N . That is, when the N rows (or N columns) are **independent**. In this case, your ability to “predict” or “calculate” each of the N observations is independent.

3.6.5 Example of Minimum Length Problem

Let's reconsider the four-parameter tomography problem we introduced in Section 3.4.4. With our new understanding of the determinacy of least squares problems, we can now recognize the problem in Section 3.4.4 as an underdetermined problem with four unknowns and three independent data. The least squares solution, \mathbf{m}_{LS} , does not exist, but the minimum length solution, \mathbf{m}_{ML} , does exist. In this case, \mathbf{G} is a 3×4 matrix. Why?

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (3.71)$$

$$\mathbf{G}\mathbf{G}^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (3.72)$$

The inverse for $\mathbf{G}\mathbf{G}^T$ exists, so we can compute

$$\mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix} \quad (3.73)$$

For an actual model $\mathbf{m} = [1 \ 0 \ 0 \ 0]$, the data are $\mathbf{d} = [1 \ 0 \ 1]$. The minimum length solution is given by

$$\mathbf{m}_{\text{ML}} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\mathbf{d} = [1 \ 0 \ 0 \ 0] \quad (3.74)$$

In this particular case, \mathbf{m}_{ML} is the correct solution. This is not always the case. In fact, in most realistic cases, it is not the “correct” solution, if one is known. Remember, this is the minimum length solution that solves the problem, but it is not unique or, for that matter, “correct.”

3.7 Weighted Measures of Length

3.7.1 Introduction

One way to improve our estimates using either the least squares solution

$$\mathbf{m}_{\text{LS}} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d} \quad (3.27)$$

or the minimum length solution

$$\mathbf{m}_{\text{ML}} = \langle \mathbf{m} \rangle + \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}[\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] \quad (3.70)$$

is to use *weighted* measures of the misfit vector

$$\mathbf{e} = \mathbf{d}^{\text{obs}} - \mathbf{d}^{\text{pre}} \quad (3.75)$$

or the model parameter vector \mathbf{m} , respectively. The next two subsections will deal with these two approaches.

3.7.2 Weighted Least Squares

Weighted Measures of the Misfit Vector e

We saw in Section 3.4 that the least squares solution \mathbf{m}_{LS} was the one that minimized the total misfit between predicted and observed data in the L_2 norm sense. That is, E in

$$E = \mathbf{e}^T\mathbf{e} = [e_1 \ e_2 \ \cdots \ e_N] \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \sum_{i=1}^N e_i^2 \quad (3.4)$$

is minimized.

Consider a new E , defined as follows:

$$E = \mathbf{e}^T\mathbf{W}_e\mathbf{e} \quad (3.76)$$

and where \mathbf{W}_e is an, as yet, unspecified $N \times N$ weighting matrix. \mathbf{W}_e can take any form, but one convenient choice is

$$\mathbf{W}_e = [\text{cov } \mathbf{d}]^{-1} \quad (3.77)$$

where $[\text{cov } \mathbf{d}]^{-1}$ is the inverse of the covariance matrix for the data. With this choice for the weighting matrix, **data with large variances are weighted less than ones with small variances**. While this is true in general, it is easier to show in the case where \mathbf{W}_e is diagonal. This happens when $[\text{cov } \mathbf{d}]$ is diagonal, **which implies that the errors in the data are uncorrelated**. The diagonal entries in $[\text{cov } \mathbf{d}]^{-1}$ are then given by the reciprocal of the diagonal entries in $[\text{cov } \mathbf{d}]$. That is, if

$$[\text{cov } \mathbf{d}] = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_N^2 \end{bmatrix} \quad (3.78)$$

then

$$[\text{cov } \mathbf{d}]^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 & \dots & 0 \\ 0 & \sigma_2^{-2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \sigma_N^{-2} \end{bmatrix} \quad (3.79)$$

With this choice for \mathbf{W}_e , the weighted misfit becomes

$$E = \mathbf{e}^T \mathbf{W}_e \mathbf{e} = \sum_{i=1}^N \left[e_i \sum_{j=1}^N W_{ij} e_j \right] \quad (3.80)$$

But,

$$W_{ij} = \delta_{ij} \frac{1}{\sigma_i^2} \quad (3.81)$$

where δ_{ij} is the Kronecker delta. Thus, we have

$$E = \sum_{i=1}^N \frac{1}{\sigma_i^2} e_i^2 \quad (3.82)$$

If the i th variance σ_i^2 is large, then the component of the error vector in the i th direction, e_i^2 , has little influence on the size of E . This is not the case in the unweighted least squares problem, where an examination of Equation (3.4) clearly shows that each component of the error vector contributes equally to the total misfit.

Obtaining the Weighted Least Squares Solution \mathbf{m}_{WLS}

If one uses $E = \mathbf{e}^T \mathbf{W}_e \mathbf{e}$ as the weighted measure of error, we will see below that this leads to the weighted least squares solution:

$$\mathbf{m}_{\text{WLS}} = [\mathbf{G}^T \mathbf{W}_e \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{W}_e \mathbf{d} \quad (3.83)$$

with a weighted least squares operator $\mathbf{G}_{\text{WLS}}^{-1}$ given by

$$\mathbf{G}_{\text{WLS}}^{-1} = [\mathbf{G}^T \mathbf{W}_e \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{W}_e \quad (3.84)$$

While this is true in general, it is easier to arrive at Equation (3.83) in the case where \mathbf{W}_e is a diagonal matrix and the forward problem $\mathbf{d} = \mathbf{G}\mathbf{m}$ is given by the least squares problem for a best-fitting straight line [see Equation (3.6)].

Step 1.

$$E = \mathbf{e}^T \mathbf{W}_e \mathbf{e} = \sum_{i=1}^N \left[e_i \sum_{j=1}^M W_{ij} e_j \right] = \sum_{i=1}^N W_{ii} e_i^2 \quad (3.85)$$

$$\begin{aligned} &= \sum_{i=1}^N W_{ii} (d_i^{\text{obs}} - d_i^{\text{pre}})^2 = \sum_{i=1}^N W_{ii} \left[d_i - \sum_{j=1}^M G_{ij} m_j \right]^2 \\ &= \sum_{i=1}^N W_{ii} (d_i^2 - 2m_1 d_i - 2m_2 d_i z_i + m_1^2 + 2m_1 m_2 z_i + m_2^2 z_i^2) \end{aligned} \quad (3.86)$$

Step 2. Then

$$\frac{\partial E}{\partial m_1} = -2 \sum_{i=1}^N d_i W_{ii} + 2m_1 \sum_{i=1}^N W_{ii} + 2m_2 \sum_{i=1}^N z_i W_{ii} = 0 \quad (3.87)$$

and

$$\frac{\partial E}{\partial m_2} = -2 \sum_{i=1}^N d_i z_i W_{ii} + 2m_1 \sum_{i=1}^N z_i W_{ii} + 2m_2 \sum_{i=1}^N z_i^2 W_{ii} = 0 \quad (3.88)$$

This can be written in matrix form as

$$\begin{bmatrix} \sum_{i=1}^N W_{ii} & \sum_{i=1}^N z_i W_{ii} \\ \sum_{i=1}^N z_i W_{ii} & \sum_{i=1}^N z_i^2 W_{ii} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N d_i W_{ii} \\ \sum_{i=1}^N z_i d_i W_{ii} \end{bmatrix} \quad (3.89)$$

Step 3. The left-hand side can be factored as

$$\begin{bmatrix} \sum W_{ii} & \sum z_i W_{ii} \\ \sum z_i W_{ii} & \sum z_i^2 W_{ii} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} W_{11} & 0 & \cdots & 0 \\ 0 & W_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_{NN} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \quad (3.90)$$

or simply

$$\begin{bmatrix} \sum W_{ii} & \sum z_i W_{ii} \\ \sum z_i W_{ii} & \sum z_i^2 W_{ii} \end{bmatrix} = \mathbf{G}^T \mathbf{W}_e \mathbf{G} \quad (3.91)$$

Similarly, the right-hand side can be factored as

$$\begin{bmatrix} \sum d_i W_{ii} \\ \sum d_i z_i W_{ii} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_N \end{bmatrix} \begin{bmatrix} W_{11} & 0 & \cdots & 0 \\ 0 & W_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_{NN} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad (3.92)$$

or simply

$$\begin{bmatrix} \sum d_i W_{ii} \\ \sum d_i z_i W_{ii} \end{bmatrix} = \mathbf{G}^T \mathbf{W}_e \mathbf{d} \quad (3.93)$$

Step 4. Therefore, using Equations (3.91) and (3.93), Equation (3.89) can be written as

$$\mathbf{G}^T \mathbf{W}_e \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{W}_e \mathbf{d} \quad (3.94)$$

The weighted least squares solution, \mathbf{m}_{WLS} from Equation (3.94) is thus

$$\mathbf{m}_{\text{WLS}} = [\mathbf{G}^T \mathbf{W}_e \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{W}_e \mathbf{d} \quad (3.95)$$

assuming that $[\mathbf{G}^T \mathbf{W}_e \mathbf{G}]^{-1}$ exists, of course.

3.7.3 Weighted Minimum Length

The development of a weighted minimum length solution is similar to that of the weighted least squares problem. The steps are as follows.

First, recall that the minimum length solution minimizes $\mathbf{m}^T \mathbf{m}$. By analogy with weighted least squares, we can choose to minimize

$$\mathbf{m}^T \mathbf{W}_m \mathbf{m} \quad (3.96)$$

instead of $\mathbf{m}^T \mathbf{m}$. For example, if one wishes to use

$$\mathbf{W}_m = [\text{cov } \mathbf{m}]^{-1} \quad (3.97)$$

then one must replace \mathbf{m} above with

$$\mathbf{m} - \langle \mathbf{m} \rangle \quad (3.98)$$

where $\langle \mathbf{m} \rangle$ is the expected, or *a priori*, estimate for the parameter values. The reason for this is that the variances must represent fluctuations about zero. In the weighted least squares problem, it is assumed that the error vector \mathbf{e} which is being minimized has a mean of zero. Thus, for the weighted minimum length problem, we replace \mathbf{m} by its departure from the expected value $\langle \mathbf{m} \rangle$. Therefore, we introduce a new function L to be minimized:

$$L = [\mathbf{m} - \langle \mathbf{m} \rangle]^T \mathbf{W}_m [\mathbf{m} - \langle \mathbf{m} \rangle] \quad (3.99)$$

If one then follows the procedure in Section 3.6 with this new function, one eventually (as in “It is left to the student as an exercise!!”) is led to the weighted minimum length solution \mathbf{m}_{WML} given by

$$\mathbf{m}_{\text{WML}} = \langle \mathbf{m} \rangle + \mathbf{W}_m^{-1} \mathbf{G}^T [\mathbf{G} \mathbf{W}_m^{-1} \mathbf{G}^T]^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \quad (3.100)$$

and the weighted minimum length operator, $\mathbf{G}_{\text{WML}}^{-1}$, is given by

$$\mathbf{G}_{\text{WML}}^{-1} = \mathbf{W}_m^{-1} \mathbf{G}^T [\mathbf{G} \mathbf{W}_m^{-1} \mathbf{G}^T]^{-1} \quad (3.101)$$

This expression differs from Equation (3.38), page 54 of Menke, which uses \mathbf{W}_m rather than \mathbf{W}_m^{-1} . I believe Menke’s equation is wrong. Note that the solution depends explicitly on the expected, or *a priori*, estimate of the model parameters $\langle \mathbf{m} \rangle$. The second term represents a departure from the *a priori* estimate $\langle \mathbf{m} \rangle$, based on the inadequacy of the forward problem $\mathbf{G} \langle \mathbf{m} \rangle$ to fit the data \mathbf{d} exactly.

Other choices for \mathbf{W}_m include:

1. $\mathbf{D}^T \mathbf{D}$, where \mathbf{D} is a derivative matrix (a measure of the flatness of \mathbf{m}) of dimension $(M - 1) \times M$:

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \quad (3.102)$$

2. $\mathbf{D}^1 \mathbf{D}$, where \mathbf{D} is an $(M - 2) \times M$ roughness (second derivative) matrix given by

$$\mathbf{D} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} \quad (3.103)$$

Note that for both choices of \mathbf{D} presented, $\mathbf{D}^T\mathbf{D}$ is an $M \times M$ matrix of rank less than M (for the first-derivative case, it is of rank $M - 1$, while for the second it is of rank $M - 2$). This means that \mathbf{W}_m does not have a mathematical inverse. This can introduce some nonuniqueness into the solution, but does not preclude finding a solution. Finally, note that many choices for \mathbf{W}_m are possible.

3.7.4 Weighted Damped Least Squares

In Sections 3.7.2 and 3.7.3 we considered weighted versions of the least squares and minimum length solutions. Both unweighted and weighted problems can be very unstable if the matrices that have to be inverted are nearly singular. In the weighted problems, these are

$$\mathbf{G}^T\mathbf{W}_e\mathbf{G} \quad (3.104)$$

and

$$\mathbf{G}\mathbf{W}_m^{-1}\mathbf{G}^T \quad (3.105)$$

respectively, for least squares and minimum length problems. In this case, one can form a weighted penalty, or cost function, given by

$$E + \varepsilon^2 L \quad (3.106)$$

where E is from Equation (3.85) for weighted least squares and L is from Equation (3.99) for the weighted minimum length problem. One then goes through the exercise of minimizing Equation (3.106) with respect to the model parameters \mathbf{m} , and obtains what is known as the weighted, damped least squares solution \mathbf{m}_{WD} . It is, in fact, a weighted mix of the weighted least squares and weighted minimum length solutions.

One finds that \mathbf{m}_{WD} is given by either

$$\mathbf{m}_{\text{WD}} = \langle \mathbf{m} \rangle + [\mathbf{G}^T\mathbf{W}_e\mathbf{G} + \varepsilon^2\mathbf{W}_m]^{-1}\mathbf{G}^T\mathbf{W}_e[\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] \quad (3.107)$$

or

$$\mathbf{m}_{\text{WD}} = \langle \mathbf{m} \rangle + \mathbf{W}_m^{-1}\mathbf{G}^T[\mathbf{G}\mathbf{W}_m^{-1}\mathbf{G}^T + \varepsilon^2\mathbf{W}_e^{-1}]^{-1}[\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle] \quad (3.108)$$

where the weighted, damped least squares operator, $\mathbf{G}_{\text{WD}}^{-1}$, is given by

$$\mathbf{G}_{\text{WD}}^{-1} = [\mathbf{G}^T\mathbf{W}_e\mathbf{G} + \varepsilon^2\mathbf{W}_m]^{-1}\mathbf{G}^T\mathbf{W}_e \quad (3.109)$$

or

$$\mathbf{G}_{\text{WD}}^{-1} = \mathbf{W}_m^{-1}\mathbf{G}^T[\mathbf{G}\mathbf{W}_m^{-1}\mathbf{G}^T + \varepsilon^2\mathbf{W}_e^{-1}]^{-1} \quad (3.110)$$

The two forms for $\mathbf{G}_{\text{WD}}^{-1}$ can be shown to be equivalent. The ε^2 term has the effect of damping the instability. As we will see later in Chapter 6 using *singular-value decomposition*, the above procedure minimizes the effects of small singular values in $\mathbf{G}^T \mathbf{W}_e \mathbf{G}$ or $\mathbf{G} \mathbf{W}_m^{-1} \mathbf{G}^T$.

In the next section we will learn two methods of including *a priori* information and constraints in inverse problems.

3.8 A Priori Information and Constraints (See Menke, Pages 55–57)

3.8.1 Introduction

Another common type of *a priori* information takes the form of *linear equality constraints*:

$$\mathbf{Fm} = \mathbf{h} \tag{3.111}$$

where \mathbf{F} is a $P \times M$ matrix, and P is the number of linear constraints considered. As an example, consider the case for which the mean of the model parameters is known. In this case with only one constraint, we have

$$\frac{1}{M} \sum_{i=1}^M m_i = h_1 \tag{3.112}$$

Then, Equation (3.111) can be written as

$$\mathbf{Fm} = \frac{1}{M} [1 \quad 1 \quad \dots \quad 1] \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} = h_1 \tag{3.113}$$

As another example, suppose that the j th model parameter m_j is actually known in advance. That is, suppose

$$m_j = h_1 \tag{3.114}$$

Then Equation (3.111) takes the form

$$\mathbf{Fm} = [0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0] \begin{bmatrix} m_1 \\ \vdots \\ m_j \\ \vdots \\ m_M \end{bmatrix} = h_1 \tag{3.115}$$

\uparrow
 j th column

Note that for this example it would be possible to remove m_j as an unknown, thereby reducing the system of equations by one. It is often preferable to use Equation (3.94), even in this case, rather than rewriting the forward problem in a computer code.

3.8.2 A First Approach to Including Constraints

We will consider two basic approaches to including constraints in inverse problems. Each has its strengths and weaknesses. The first includes the constraint matrix \mathbf{F} in the forward problem, and the second uses Lagrange multipliers. The steps for the first approach are as follows.

Step 1. Include $\mathbf{Fm} = \mathbf{h}$ as rows in a new \mathbf{G} that operates on the original \mathbf{m} :

$$\begin{array}{ccc} \begin{bmatrix} \mathbf{G} \\ \mathbf{F} \end{bmatrix} & \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} & = \begin{bmatrix} \mathbf{d} \\ \mathbf{h} \end{bmatrix} \\ (N+P) \times M & M \times 1 & (N+P) \times 1 \end{array} \quad (3.116)$$

Step 2. The new $(N+P) \times 1$ misfit vector \mathbf{e} becomes

$$\mathbf{e} = \begin{array}{c} \begin{bmatrix} \mathbf{d}^{\text{obs}} \\ \mathbf{h} \end{bmatrix} \\ (N+P) \times 1 \end{array} - \begin{array}{c} \begin{bmatrix} \mathbf{d}^{\text{pre}} \\ \mathbf{h}^{\text{pre}} \end{bmatrix} \\ (N+P) \times 1 \end{array} \quad (3.117)$$

Performing a least squares inversion would minimize the new $\mathbf{e}^T \mathbf{e}$, based on Equation (3.116). The difference

$$\mathbf{h} - \mathbf{h}^{\text{pre}} \quad (3.118)$$

which represents the misfit to the constraints, may be small, but it is unlikely that it would vanish, which it must if the constraints are to be satisfied.

Step 3. Introduce a weighted misfit:

$$\mathbf{e}^T \mathbf{W}_e \mathbf{e} \quad (3.119)$$

where \mathbf{W}_e is a diagonal matrix of the form

$$W_e = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & \\ & & 1 & \vdots \\ \vdots & & (\text{big \#}) & \\ 0 & \dots & 0 & (\text{big \#}) \end{bmatrix} \begin{matrix} \uparrow \\ N \\ \downarrow \\ \uparrow \\ P \\ \downarrow \end{matrix} \quad (3.120)$$

That is, it has relatively large values for the last P entries associated with the constraint equations. Recalling the form of the weighting matrix used in Equation (3.77), one sees that Equation (3.120) is equivalent to assigning the constraints very small variances. Hence, a weighted least squares approach in this case will give large weight to fitting the constraints. The size of the big numbers in \mathbf{W}_e must be determined empirically. One seeks a number that leads to a solution that satisfies the constraints acceptably, but does not make the matrix in Equation (3.104) that must be inverted to obtain the solution too poorly conditioned. Matrices with a large range of values in them tend to be poorly conditioned.

Consider the example of the smoothing constraint here, $P = M - 2$:

$$\mathbf{D}\mathbf{m} = \mathbf{0} \quad (3.121)$$

where the dimensions of $\mathbf{D} = (M - 2) \times m$, $\mathbf{m} = M \times 1$, and $\mathbf{0} = (M - 2) \times 1$. The augmented equations are

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \quad (3.122)$$

Let's use the following weighting matrix:

$$W_e = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \theta^2 & 0 \\ 0 & \dots & \dots & 0 & \theta^2 \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{I}_{N \times N} & \mathbf{0} \\ \hline \mathbf{0} & \theta^2 \mathbf{I}_{P \times P} \end{array} \right] \quad (3.123)$$

where θ^2 is a constant. This results in the following, with the dimensions of the three matrices in the first set of brackets being $M \times (N + P)$, $(N + P) \times (N + P)$, and $(N + P) \times M$.

$$\mathbf{m}_{\text{WLS}} = \left\{ \begin{bmatrix} \mathbf{G} \\ \mathbf{D} \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \theta^2 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \mathbf{D} \end{bmatrix} \right\}^{-1} \begin{bmatrix} \mathbf{G} \\ \mathbf{D} \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \theta^2 \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}$$

$$\begin{matrix} \xleftarrow{\hspace{10em}} & \xleftarrow{\hspace{10em}} \\ \left[\mathbf{G}^T \mid \theta^2 \mathbf{D}^T \right] & \left[\mathbf{G}^T \mid \theta^2 \mathbf{D}^T \right] \end{matrix}$$

The lower matrices having dimensions of $M \times (N+P) \mid (N+P) \times 1$.

$$= \begin{bmatrix} \mathbf{G}^T \mathbf{G} + \theta^2 \mathbf{D}^T \mathbf{D} & \\ & \mathbf{G}^T \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{G}^T \mathbf{D} \mathbf{d} \\ \mathbf{G}^T \mathbf{d} \end{bmatrix} \quad (3.124)$$

$$= \left\{ \begin{bmatrix} \mathbf{G} \\ \theta \mathbf{D} \end{bmatrix}^T \begin{bmatrix} \mathbf{G} \\ \theta \mathbf{D} \end{bmatrix} \right\}^{-1} \begin{bmatrix} \mathbf{G}^T \mathbf{d} \\ \mathbf{G}^T \mathbf{d} \end{bmatrix} \quad (3.125)$$

The three matrices within (3.125) have dimensions $M \times (N+P)$, $(N+P) \times M$, and $M \times 1$, which produce an $M \times 1$ matrix when evaluated. In this form we can see this is simply the \mathbf{m}_{LS} for the problem

$$\begin{bmatrix} \mathbf{G} \\ \theta \mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \quad (3.126)$$

By varying θ , we can trade off the misfit and the smoothness for the model.

3.8.3 A Second Approach to Including Constraints

Whenever the subject of constraints is raised, Lagrange multipliers come to mind! The steps for this approach are as follows.

Step 1. Form a weighted sum of the misfit and the constraints:

$$\phi(\mathbf{m}) = \mathbf{e}^T \mathbf{e} + [\mathbf{Fm} - \mathbf{h}]^T \boldsymbol{\lambda} \quad (3.127)$$

which can be expanded as

$$\phi(\mathbf{m}) = \sum_{i=1}^N \left[d_i - \sum_{j=1}^M G_{ij} m_j \right]^2 + 2 \sum_{i=1}^P \lambda_i \left[\sum_{j=1}^M F_{ij} m_j - h_i \right] \quad (3.128)$$

where \uparrow indicates a difference from Equation (3.43) on page 56 in Menke, and where there are P linear equality constraints and where the factor of 2 as been added as a matter of convenience to make the form of the final answer simpler.

Step 2. One then takes the partials of Equation (3.128) with respect to all the entries in \mathbf{m} and sets them to zero. That is,

$$\frac{\partial \phi(\mathbf{m})}{\partial m_q} = 0 \quad q = 1, 2, \dots, M \quad (3.129)$$

which leads to

$$2 \sum_{i=1}^M m_i \sum_{j=1}^N G_{jq} G_{ji} - 2 \sum_{i=1}^N G_{iq} d_i + 2 \sum_{i=1}^P \lambda_i F_{iq} = 0 \quad q = 1, 2, \dots, M \quad (3.130)$$

where the first two terms are the same as the least squares case in Equation (3.22a) since they come directly from $\mathbf{e}^T \mathbf{e}$ and the last term shows why the factor of 2 was added in Equation (3.128).

Step 3. Equation (3.130) is not the complete description of the problem. To the M equations in Equation (3.130), P constraint equations must also be added. In matrix form, this yields

$$\begin{bmatrix} \mathbf{G}^T \mathbf{G} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^T \mathbf{d} \\ \mathbf{h} \end{bmatrix} \quad (3.131)$$

$(M + P) \times (M + P) \quad (M + P) \times 1 \quad (M + P) \times 1$

Step 4. The above system of equations can be solved as

$$\begin{bmatrix} \mathbf{m} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^T \mathbf{G} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{G}^T \mathbf{d} \\ \mathbf{h} \end{bmatrix} \quad (3.132)$$

As an example, consider constraining a straight line to pass through some point (z' , d'). That is, for N observations, we have

$$d_i = m_1 + m_2 z_i \quad i = 1, N \quad (3.133)$$

subject to the single constraint

$$d' = m_1 + m_2 z' \quad (3.134)$$

Then Equation (3.111) has the form

$$\mathbf{Fm} = \begin{bmatrix} 1 & z' \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = d' \quad (3.135)$$

We can then write out Equation (3.132) explicitly, and obtain the following:

$$\begin{bmatrix} m_1 \\ m_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} N & \Sigma z_i & 1 \\ \Sigma z_i & \Sigma z_i^2 & z' \\ 1 & z' & 0 \end{bmatrix}^{-1} \begin{bmatrix} \Sigma d_i \\ \Sigma z_i d_i \\ d' \end{bmatrix} \quad (3.136)$$

Note the similarity between Equations (3.136) and (3.32), the least squares solution to fitting a straight line to a set of points without any constraints:

$$\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}_{\text{LS}} = \begin{bmatrix} N & \sum z_i \\ \sum z_i & \sum z_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum d_i \\ \sum z_i d_i \end{bmatrix} \quad (3.32)$$

If you wanted to get the same result for the straight line passing through a point using the first approach with \mathbf{W}_e , you would assign

$$W_{ii} = 1 \quad i = 1, \dots, N \quad (3.137)$$

and

$$W_{N+1, N+1} = \text{big \#} \quad (3.138)$$

which is equivalent to assigning a small variance (relative to the unconstrained part of the problem) to the constraint equation. The solution obtained with Equation (3.96) should approach the solution obtained using Equation (3.136).

Note that it is easy to constrain lines to pass through the origin using Equation (3.136). In this case, we have

$$d' = z' = 0 \quad (3.139)$$

and Equation (3.136) becomes

$$\begin{bmatrix} m_1 \\ m_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} N & \sum z_i & 1 \\ \sum z_i & \sum z_i^2 & 0 \\ 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \sum d_i \\ \sum z_i d_i \\ 0 \end{bmatrix} \quad (3.140)$$

The advantage of using the Lagrange multiplier approach to constraints is that the constraints will be satisfied exactly. It often happens, however, that the constraints are only approximately known, and using Lagrange multipliers to fit the constraints exactly may not be appropriate. An example might be a gravity inversion where depth to bedrock at one point is known from drilling. Constraining the depth to be exactly the drill depth may be misleading if the depth in the model is an average over some area. Then the exact depth at one point may not be the best estimate of the depth over the area in question. A second disadvantage of the Lagrange multiplier approach is that it adds one equation to the system of equations in Equation (3.136) for each constraint. This can add up quickly, making the inversion considerably more difficult computationally.

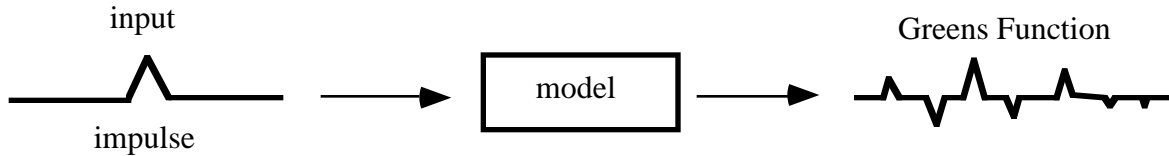
An entirely different class of constraints are called *linear inequality constraints* and take the form

$$\mathbf{Fm} \geq \mathbf{h} \quad (3.141)$$

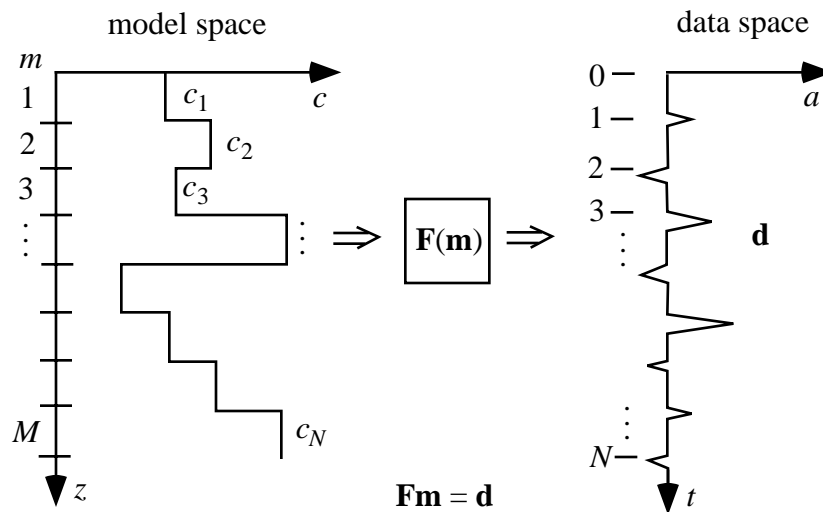
These can be solved using *linear programming* techniques, but we will not consider them further in this class.

3.8.4 Seismic Receiver Function Example

The following is an example of using smoothing constraints in an inverse problem. Consider a general problem in time series analysis, with a delta function input. Then the output from the "model" is the *Greens function* of the system. The inverse problem is this: Given the Greens function, find the parameters of the model.



In a little more concrete form:



If \mathbf{d} is very noisy, then \mathbf{m}_{LS} will have a high-frequency component to try to "fit the noise," but this will not be real. How do we prevent this? So far, we have learned two ways: use \mathbf{m}_{WLS} if we know $\text{cov } \mathbf{d}$, or if not, we can place a smoothing constraint on \mathbf{m} . An example of this approach using receiver function inversions can be found in *Ammon et al.* (1990).

The important points are as follows:

- This approach is used in the real world.
- The forward problem is written

$$d_j = F_j \mathbf{m} \quad j = 1, 2, 3 \dots N$$

- This is nonlinear, but after linearization (discussed in Chapter 4), the equations are the same as discussed previously (with minor differences).
- Note the correlation between the roughness in the model and the roughness in the data.
- The way to choose the weighting parameter, σ , is to plot the trade-off between smoothness and waveform fit.

3.9 Variances of Model Parameters (See Pages 58–60, Menke)

3.9.1 Introduction

Data errors are mapped into model parameter errors through any type of inverse. We noted in Chapter 2 [Equations (2.41)–(2.43)] that if

$$\mathbf{m}^{\text{est}} = \mathbf{M}\mathbf{d} + \mathbf{v} \quad (2.41)$$

and if $[\text{cov } \mathbf{d}]$ is the data covariance matrix which describes the data errors, then the *a posteriori* model covariance matrix is given by

$$[\text{cov } \mathbf{m}] = \mathbf{M}[\text{cov } \mathbf{d}]\mathbf{M}^T \quad (2.44c)$$

The covariance matrix in Equation (2.44c) is called the *a posteriori model covariance matrix* because it is calculated after the fact. It gives what are sometimes called the formal uncertainties in the model parameters. It is different from the *a priori* model covariance matrix of Equation (3.79), which is used to constrain the underdetermined problem.

The *a posteriori* covariance matrix in Equation (2.44c) shows explicitly the mapping of data errors into uncertainties in the model parameters. Although the mapping will be clearer once we consider the generalized inverse in Chapter 7, it is instructive at this point to consider applying Equation (2.44c) to the least squares and minimum length problems.

3.9.2 Application to Least Squares

We can apply Equation (2.44c) to the least squares problem and obtain

$$[\text{cov } \mathbf{m}] = \{[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\}[\text{cov } \mathbf{d}]\{[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\}^T \quad (3.142)$$

Further, if $[\text{cov } \mathbf{d}]$ is given by

$$[\text{cov } \mathbf{d}] = \sigma^2\mathbf{I}_N \quad (3.143)$$

then

$$\begin{aligned} [\text{cov } \mathbf{m}] &= [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T[\sigma^2\mathbf{I}]\{[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\}^T \\ &= \sigma^2 [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{G}\{[\mathbf{G}^T\mathbf{G}]^{-1}\}^T \\ &= \sigma^2 \{[\mathbf{G}^T\mathbf{G}]^{-1}\}^T \\ &= \sigma^2 [\mathbf{G}^T\mathbf{G}]^{-1} \end{aligned} \quad (3.144)$$

since the transpose of a symmetric matrix returns the original matrix.

3.9.3 Application to the Minimum Length Problem

Application of Equation (2.44c) to the minimum length problem leads to the following for the *a posteriori* model covariance matrix:

$$[\text{cov } \mathbf{m}] = \{\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\}[\text{cov } \mathbf{d}]\{\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\}^T \quad (3.145)$$

If the data covariance matrix is again given by

$$[\text{cov } \mathbf{d}] = \sigma^2 \mathbf{I}_N \quad (3.146)$$

we obtain

$$[\text{cov } \mathbf{m}] = \sigma^2 \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-2}\mathbf{G} \quad (3.147)$$

where

$$[\mathbf{G}\mathbf{G}^T]^{-2} = [\mathbf{G}\mathbf{G}^T]^{-1}[\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.148)$$

3.9.4 Geometrical Interpretation of Variance

There is another way to look at the variance of model parameter estimates for the least squares problem that considers the prediction error, or misfit, to the data. Recall that we defined the misfit E as

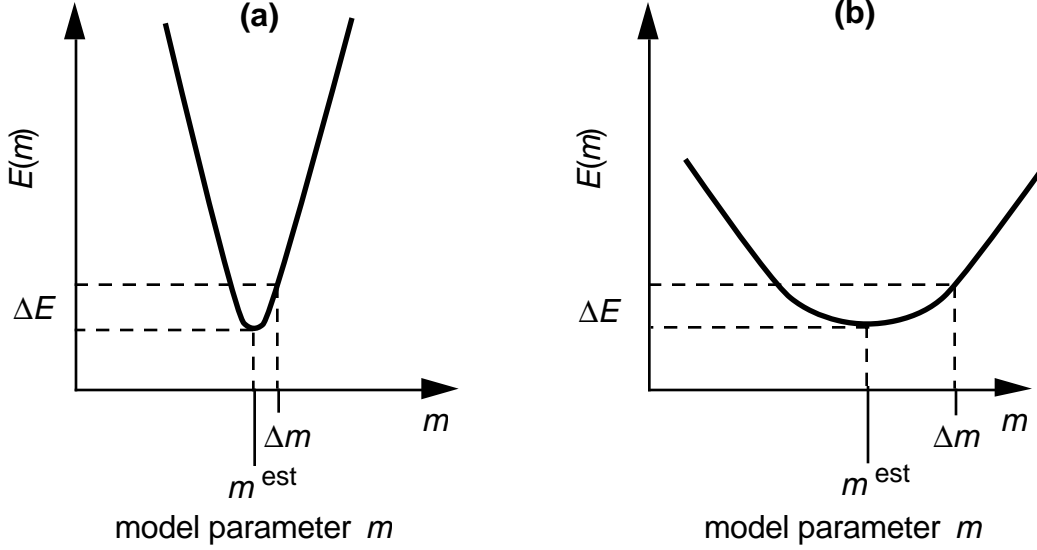
$$\begin{aligned} E &= \mathbf{e}^T \mathbf{e} = [\mathbf{d} - \mathbf{d}^{\text{pre}}]^T [\mathbf{d} - \mathbf{d}^{\text{pre}}] \\ &= [\mathbf{d} - \mathbf{G}\mathbf{m}]^T [\mathbf{d} - \mathbf{G}\mathbf{m}] \end{aligned} \quad (3.20)$$

which explicitly shows the dependence of E on the model parameters \mathbf{m} . That is, we have

$$E = E(\mathbf{m}) \quad (3.149)$$

If $E(\mathbf{m})$ has a sharp, well-defined minimum, then we can conclude that our solution \mathbf{m}_{LS} is well constrained. Conversely, if $E(\mathbf{m})$ has a broad, poorly defined minimum, then we conclude that our solution \mathbf{m}_{LS} is poorly constrained.

After Figure 3.10, page 59, of Menke, we have (next page),



(a) The best estimate m^{est} of model parameter m occurs at the minimum of $E(m)$. If the minimum is relatively narrow, then random fluctuations in $E(m)$ lead to only small errors Δm in m^{est} . (b) If the minimum is wide, then large errors in m can occur.

One way to quantify this qualitative observation is to realize that the width of the minimum for $E(\mathbf{m})$ is related to the curvature, or second derivative, of $E(\mathbf{m})$ at the minimum. For the least squares problem, we have

$$\left. \frac{\partial^2 E}{\partial \mathbf{m}^2} \right|_{\mathbf{m}=\mathbf{m}_{\text{LS}}} = \left. \frac{\partial^2}{\partial \mathbf{m}^2} [\mathbf{d} - \mathbf{Gm}]^2 \right|_{\mathbf{m}=\mathbf{m}_{\text{LS}}} \quad (3.150)$$

Evaluating the right-hand side, we have for the q th term

$$\frac{\partial^2 E}{\partial m_q^2} = \frac{\partial^2}{\partial m_q^2} [\mathbf{d} - \mathbf{Gm}]^2 = \frac{\partial^2}{\partial m_q^2} \sum_{i=1}^N \left[d_i - \sum_{j=1}^M G_{ij} m_j \right]^2 \quad (3.151)$$

$$= \frac{\partial^2}{\partial m_q^2} 2 \sum_{i=1}^N \left[d_i - \sum_{j=1}^M G_{ij} m_j \right] (-G_{iq}) \quad (3.152)$$

$$= -2 \frac{\partial^2}{\partial m_q^2} \sum_{i=1}^N \left[G_{iq} d_i - \sum_{j=1}^M G_{ij} G_{iq} m_j \right] \quad (3.153)$$

Using the same steps as we did in the derivation of the least squares solution in Equations (3.21)–(3.25), it is possible to see that Equation (3.153) represents the q th term in $\mathbf{G}^T[\mathbf{d} - \mathbf{Gm}]$. Combining the q equations into matrix notation yields

$$\frac{\partial^2}{\partial \mathbf{m}^2} [\mathbf{d} - \mathbf{G}\mathbf{m}]^2 = -2 \frac{\partial}{\partial \mathbf{m}} \{ \mathbf{G}^T [\mathbf{d} - \mathbf{G}\mathbf{m}] \} \quad (3.154)$$

Evaluating the first derivative on the right-hand side of Equation (3.154), we have for the q th term

$$\frac{\partial}{\partial m_q} \{ \mathbf{G}^T [\mathbf{d} - \mathbf{G}\mathbf{m}] \} = \frac{\partial}{\partial m_q} \sum_{i=1}^N \left[G_{iq} d_i - \sum_{j=1}^M G_{ij} G_{iq} m_j \right] \quad (3.155)$$

$$= - \sum_{i=1}^N \sum_{j=1}^M \frac{\partial}{\partial m_q} (G_{ij} G_{iq} m_j) \quad (3.156)$$

$$= - \sum_{i=1}^N G_{iq} G_{iq} \quad (3.157)$$

which we recognize as the (q, q) entry in $\mathbf{G}^T \mathbf{G}$. Therefore, we can write the $M \times M$ matrix equation as

$$\frac{\partial}{\partial \mathbf{m}} \{ \mathbf{G}^T [\mathbf{d} - \mathbf{G}\mathbf{m}] \} = -\mathbf{G}^T \mathbf{G} \quad (3.158)$$

From Equations (3.143)–(3.151) we can conclude that the second derivative of E in the least squares problem is proportional to $\mathbf{G}^T \mathbf{G}$. That is,

$$\left. \frac{\partial^2 E}{\partial \mathbf{m}^2} \right|_{\mathbf{m}=\mathbf{m}_{LS}} = (\text{constant}) \mathbf{G}^T \mathbf{G} \quad (3.159)$$

Furthermore, from Equation (3.143) we have that $[\text{cov } \mathbf{m}]$ is proportional to $[\mathbf{G}^T \mathbf{G}]^{-1}$. Therefore, we can associate large values of the second derivative of E , given by (3.159) with (1) “sharp” curvature for E , (2) “narrow” well for E , and (3) “good” (i.e., small) model variance.

As Menke points out, $[\text{cov } \mathbf{m}]$ can be interpreted as being controlled either by (1) the variance of the data times a measure of how error in the data is mapped into model parameters or (2) a constant times the curvature of the prediction error at its minimum.

I like Menke’s summary for his chapter (page 60) on this material very much. Hence, I’ve reproduced his closing paragraph for you as follows:

The methods of solving inverse problems that have been discussed in this chapter emphasize the data and model parameters themselves. The method of least squares estimates the model parameters with smallest prediction length. The method of minimum length estimates the simplest model parameters. The ideas of data and model parameters are very concrete and straightforward, and the methods based on them are simple and easily understood. Nevertheless, this viewpoint tends to obscure an important aspect of inverse problems. Namely, that the nature of the problem depends more on the *relationship* between the data and model parameters than on the data or model parameters themselves. It should, for instance, be possible

to tell a well-designed experiment from a poorly designed one without knowing what the numerical values of the data or model parameters are, or even the range in which they fall.

Before considering the relationships implied in the mapping between model parameters and data in Chapter 5, we extend what we now know about linear inverse problems to nonlinear problems in the next chapter.

CHAPTER 4: LINEARIZATION OF NONLINEAR PROBLEMS

4.1 Introduction

Thus far we have dealt with the linear, explicit forward problem given by

$$\mathbf{G}\mathbf{m} = \mathbf{d} \quad (1.13)$$

where \mathbf{G} is a matrix of coefficients (constants) that multiply the model parameter vector \mathbf{m} and return a data vector \mathbf{d} . If \mathbf{m} is doubled, then \mathbf{d} is also doubled.

We can also write Equation (1.13) out explicitly as

$$d_i = \sum_{j=1}^M G_{ij}m_j \quad i = 1, 2, \dots, N \quad (4.1)$$

This form emphasizes the linear nature of the problem. Next, we consider a more general relationship between data and model parameters.

4.2 Linearization of Nonlinear Problems

Consider a general (explicit) relationship between the i th datum and the model parameters given by

$$d_i = g_i(\mathbf{m}) \quad (4.2)$$

An example might be

$$d_1 = 2m_1^3 \quad (4.3)$$

The steps required to linearize a problem of the form of Equation (4.2) are as follows:

Step 1. Expand $g_i(\mathbf{m})$ about some point \mathbf{m}_0 in model space using a Taylor series expansion:

$$d_i = g_i(\mathbf{m}) \approx g_i(\mathbf{m}_0) + \sum_{j=1}^M \left[\frac{\partial g_i(\mathbf{m})}{\partial m_j} \Big|_{\mathbf{m}=\mathbf{m}_0} \cdot \Delta m_j \right] + \frac{1}{2} \sum_{j=1}^M \left[\frac{\partial^2 g_i(\mathbf{m})}{\partial m_j^2} \Big|_{\mathbf{m}=\mathbf{m}_0} \cdot \Delta m_j^2 \right] + 0(\Delta m_j^3) \quad (4.4)$$

where $\Delta \mathbf{m}$ is the difference between \mathbf{m} and \mathbf{m}_0 , or

$$\Delta \mathbf{m} = \mathbf{m} - \mathbf{m}_0 \quad (4.5)$$

If we assume that terms in Δm_j^n , $n \geq 2$, are small with respect to Δm_j terms, then

$$d_i = g_i(\mathbf{m}) \approx g_i(\mathbf{m}_0) + \sum_{j=1}^M \left[\frac{\partial g_i(\mathbf{m})}{\partial m_j} \Big|_{\mathbf{m}=\mathbf{m}_0} \cdot \Delta m_j \right] \quad (4.6)$$

Step 2. The predicted data \hat{d}_i at $\mathbf{m} = \mathbf{m}_0$ are given by

$$\hat{d}_i = g_i(\mathbf{m}_0) \quad (4.7)$$

Therefore

$$d_i - \hat{d}_i \approx \sum_{j=1}^M \left[\frac{\partial g_i(\mathbf{m})}{\partial m_j} \Big|_{\mathbf{m}=\mathbf{m}_0} \cdot \Delta m_j \right] \quad (4.8)$$

Step 3. We can define the misfit Δc_i as

$$\begin{aligned} \Delta c_i &= d_i - \hat{d}_i \\ &= \text{observed data} - \text{predicted data} \end{aligned} \quad (4.9)$$

Δc_i is *not necessarily* noise. It is just the misfit between observed and predicted data for some choice of the model parameter vector \mathbf{m}_0 .

Step 4. The partial derivative of the i th data equation with respect to the j th model parameter is given by

$$\frac{\partial g_i(\mathbf{m})}{\partial m_j}$$

These partial derivatives are *functions* of the model parameters and may be nonlinear (gasp) or occasionally even nonexistent (shudder).

Fortunately, the values of these partial derivatives, evaluated at some point in model space \mathbf{m}_0 , and given by

$$\frac{\partial g_i(\mathbf{m})}{\partial m_j} \Big|_{\mathbf{m}=\mathbf{m}_0} \quad (4.10b)$$

are just numbers (constants), if they exist, and not functions. We then define G_{ij} as follows:

$$G_{ij} = \left. \frac{\partial g_i(\mathbf{m})}{\partial m_j} \right|_{\mathbf{m}=\mathbf{m}_0} \quad (4.11)$$

Step 5. Finally, combining the above we have

$$\Delta c_i = \sum_{j=1}^M G_{ij} \Delta m_j \Big|_{\mathbf{m}=\mathbf{m}_0} \quad i = 1, \dots, N \quad (4.12)$$

or, in matrix notation, the linearized problem becomes

$$\Delta \mathbf{c} = \mathbf{G} \Delta \mathbf{m} \quad (4.13)$$

where

$$\begin{aligned} \Delta c_i &= d_i - \hat{d}_i = \text{observed data} - \text{predicted data} \\ &= d_i - g_i(\mathbf{m}_0) \end{aligned} \quad (4.14)$$

$$G_{ij} = \left. \frac{\partial g_i(\mathbf{m})}{\partial m_j} \right|_{\mathbf{m}=\mathbf{m}_0} \quad (4.15)$$

and

$$\Delta m_j = \text{change from } (\mathbf{m}_0)_j \quad (4.16)$$

Thus, by linearizing Equation (4.2), we have arrived at a set of linear equations, where now Δc_i (the difference between observed and predicted data) is a linear function of changes in the model parameters from some starting model.

Some general comments on Equation (4.13):

1. In general, Equation (4.13) only holds in the neighborhood of \mathbf{m}_0 , and for small changes $\Delta \mathbf{m}$. The region where the linearization is valid depends on the smoothness of $g_i(\mathbf{m})$.
2. Note that \mathbf{G} now changes with each iteration. That is, one may obtain a different \mathbf{G} for each spot in solution space. Having to reform \mathbf{G} at each step can be very time (computer) intensive, and often one uses the same \mathbf{G} for more than one iteration.

4.3 General Procedure for Nonlinear Problems

Step 1. Pick some starting model vector \mathbf{m}_0 .

Step 2. Calculate the predicted data vector $\hat{\mathbf{d}}$ and form the misfit vector

$$\Delta\mathbf{c} = \mathbf{d} - \hat{\mathbf{d}}$$

Step 3. Form

$$G_{ij} = \left. \frac{\partial g_i(\mathbf{m})}{\partial m_j} \right|_{\mathbf{m}=\mathbf{m}_0}$$

Step 4. Solve for $\Delta\mathbf{m}$ using any appropriate inverse operator (i.e., least squares, minimum length, weighted least squares, etc.)

Step 5. Form a new model parameter vector

$$\mathbf{m}_1 = \mathbf{m}_0 + \Delta\mathbf{m} \quad (4.17)$$

One repeats Steps 1–5 until $\Delta\mathbf{m}$ becomes sufficiently small (convergence is obtained) or *until* $\Delta\mathbf{c}$ becomes sufficiently small (acceptable misfit). Note that \mathbf{m}_i (note the boldfaced \mathbf{m}) is the estimate of the model parameters at the i th iteration, and not the i th component on the model parameter vector.

4.4 Three Examples

4.4.1 A Linear Example

Suppose $g_i(\mathbf{m}) = d_i$ is linear and of the form

$$2m_1 = 4 \quad (4.18)$$

With only one equation, we have $\mathbf{G} = [2]$, $\mathbf{m} = [m_1]$, and $\mathbf{d} = [4]$. (I know, I know. It's easy!) Then

$$\partial d_1 / \partial m_1 = G_{11} = 2 \quad (\text{for all } m_1)$$

Suppose that the initial estimate of the model vector $\mathbf{m}_0 = [0]$. Then $\hat{\mathbf{d}} = \mathbf{G}\mathbf{m}_0 = [2][0] = [0]$ and we have

$$\Delta\mathbf{c} = \mathbf{d} - \hat{\mathbf{d}} = [4] - [0] = [4]$$

or the change in the first and only element of our misfit vector $\Delta c_1 = 4$. Looking at our lone equation then,

$$G_{11} \Delta m_1 = \Delta c_1$$

or $2\Delta m_1 = 4$

or $\Delta m_1 = 2$

Since this is the only element in our model-change vector [in this case, $(\Delta \mathbf{m}_1)_1 = \Delta m_1$], we have $\Delta \mathbf{m}_1 = [2]$, and our next approximation of the model vector, \mathbf{m}_1 , then becomes

$$\mathbf{m}_1 = \mathbf{m}_0 + \Delta \mathbf{m}_1 = [0] + [2] = [2]$$

We have just completed Steps 1–5 for the first iteration. Now it is time to update the misfit vector and see if we have reached a solution. Thus, for the predicted data we obtain

$$\hat{\mathbf{d}} = \mathbf{G}\mathbf{m}_1 = [2][2] = [4]$$

and for the misfit we have

$$\Delta \mathbf{c} = \mathbf{d} - \hat{\mathbf{d}} = [4] - [4] = [0]$$

which indicates that the solution has converged in one iteration. To see that the solution does not depend on the starting point if Equation (4.2) is linear, let's start with

$$(\mathbf{m}_0)_1 = 1000 = m_1$$

Considering the one and only element of our predicted-data and misfit vectors, we have

$$\hat{d}_1 = 2 \times 1000 = 2000$$

and $\Delta c_1 = 4 - 2000 = -1996$

then $2\Delta m_1 = -1996$

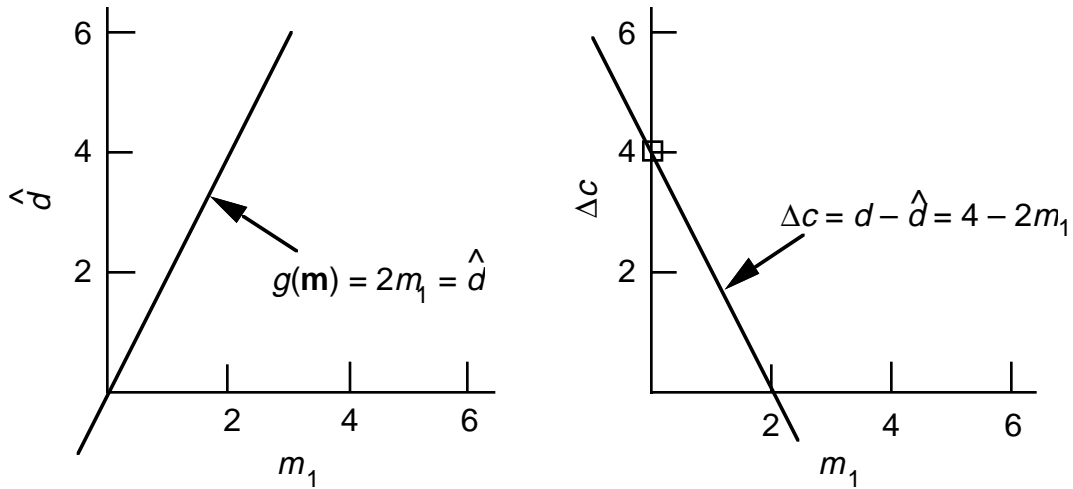
or $\Delta m_1 = -998$

Since Δm_1 is the only element of our first model-change vector, $\Delta \mathbf{m}_1$, we have $\Delta \mathbf{m}_1 = [-998]$, and therefore

$$\mathbf{m}_1 = \mathbf{m}_0 + \Delta \mathbf{m}_1 = [1000] + [-998] = [2]$$

As before, the solution has converged in one iteration. This is a general conclusion if the relationship between the data and the model parameters is linear. This problem also illustrates that the nonlinear approach outlined above works when $g_i(\mathbf{m})$ is linear.

Consider the following graphs for the system $2m_1 = 4$:



We note the following:

1. For \hat{d} versus m_1 , we see that the slope $\partial g(m_1)/\partial m_1 = 2$ for all m_1 .
2. For our initial guess of $(\mathbf{m}_0)_1 = 0$, $\Delta c = 4 - 0 = 4$, denoted by a square symbol on the plot of Δc versus m_1 . We extrapolate back down the slope to the point (m_1) where $\Delta c = 0$ to obtain our answer.
3. Because the slope does not change (the problem is linear), the starting guess \mathbf{m}_0 has no effect on the final solution. We can always get to $\Delta c = 0$ in one iteration.

4.4.2 A Nonlinear Example

Now consider, as a second example of the form $g_1(\mathbf{m}) = d_1$, the following:

$$2m^3 = 16 \tag{4.19}$$

Since we have only one unknown, I chose to drop the subscript. Instead, I will use the subscript to denote the iteration number. For example, m_3 will be the estimate of the model parameter m at the third iteration. Note also that, by inspection, $m = 2$ is the solution.

Working through this example as we did the last one, we first note that G_{11} , at the i th iteration, will be given by

$$G_{11} = \left. \frac{\partial g(m)}{\partial m} \right|_{m=m_i} = 3 \times 2m_i^2 = 6m_i^2$$

Note also that G_{11} is now a function of m .

Iteration 1. Let us pick as our starting model

$$m_0 = 1$$

then

$$G_{11} = \left. \frac{\partial g(m)}{\partial m} \right|_{m=m_0} = 6m_0^2 = 6$$

also

$$\hat{d} = 2 \times 1^3 = 2$$

and

$$\Delta c = d - \hat{d} = 16 - 2 = 14$$

Because we have only one element in our model change vector, we have $\Delta c = [14]$, and the length squared of Δc , $\|\Delta c\|^2$, is given simply by $(\Delta c)^2$:

$$(\Delta c)^2 = 14 \times 14 = 196$$

Now, we find Δm_1 , the change to m_0 , as

$$6\Delta m_1 = 14$$

and

$$\Delta m_1 = 14/6 = 2.3333$$

Thus, our estimate of the model parameter at the first iteration, m_1 , is given by

$$m_1 = m_0 + \Delta m_1 = 1 + 2.3333 = 3.3333$$

Iteration 2. Continuing,

$$G_{11} = 6m_1^2 = 66.66$$

and

$$\hat{d} = 2(3.333)^3 = 74.07$$

thus

$$\Delta c = d - \hat{d} = 16 - 74.07 = -58.07$$

now

$$(\Delta c)^2 = 3372$$

and

$$66.66\Delta m_2 = -58.07$$

gives

$$\Delta m_2 = -0.871$$

thus

$$m_2 = m_1 + \Delta m_2 = 3.3333 - 0.871 = 2.462$$

Iteration 3. Continuing,

$$G_{11} = 6m_2^2 = 36.37$$

and

$$\hat{d} = 29.847$$

thus

$$\Delta c = -13.847$$

now $(\Delta c)^2 = 192$
 and $36.37\Delta m_3 = -13.847$
 gives $\Delta m_3 = -0.381$
 thus $m_3 = m_2 + \Delta m_3 = 2.462 - 0.381 = 2.081$

Iteration 4. (Will this thing ever end??)

$G_{11} = 6m_3^2 = 25.983$
 and $\hat{d} = 18.024$
 thus $\Delta c = -2.024$
 now $(\Delta c)^2 = 4.1$
 and $25.983\Delta m_4 = -2.024$
 gives $\Delta m_4 = -0.078$
 thus $m_4 = m_3 + \Delta m_4 = 2.081 - 0.078 = 2.003$

Iteration 5. (When were computers invented???)

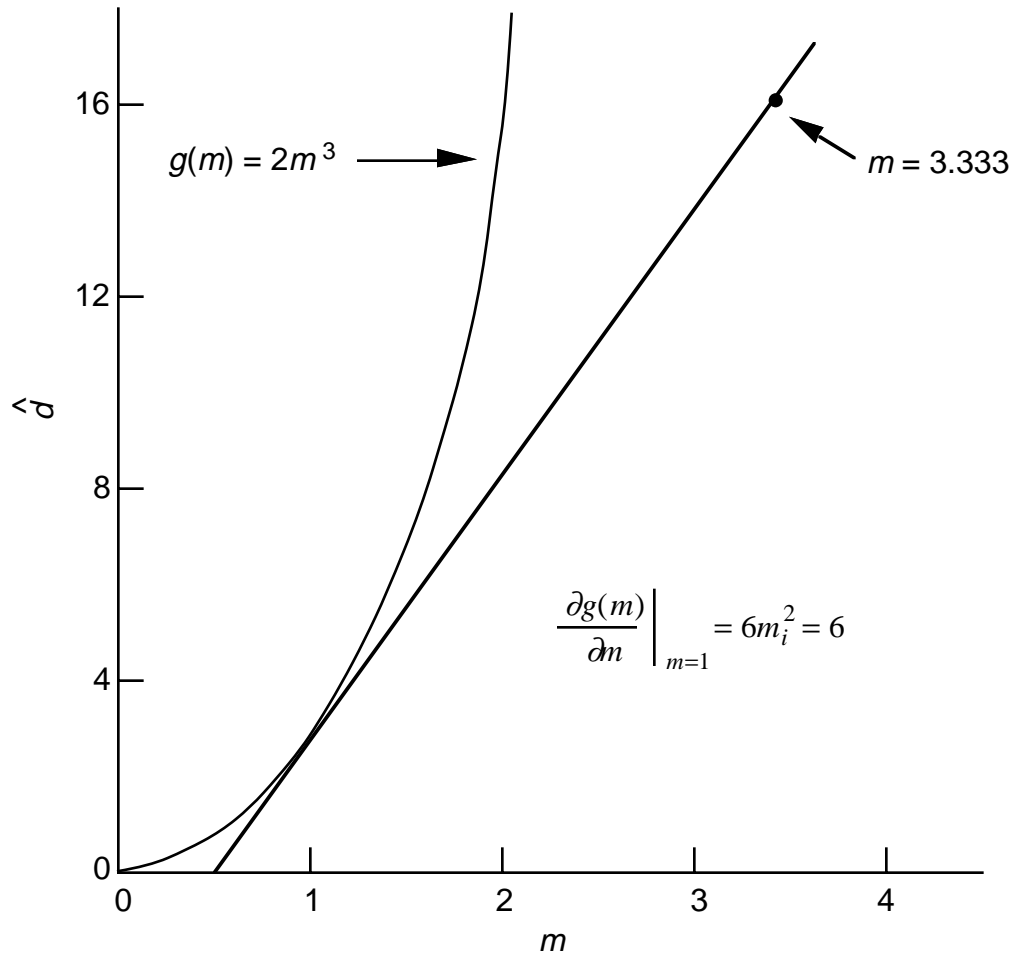
$G_{11} = 6m_4^2 = 24.072$
 and $\hat{d} = 16.072$
 thus $\Delta c = -0.072$
 now $(\Delta c)^2 = 0.005$
 and $24.072\Delta m_5 = -0.072$
 gives $\Delta m_5 = -0.003$
 thus $m_5 = m_4 + \Delta m_5 = 2.003 - 0.003 = 2.000$

Iteration 6. Beginning, we have

$G_{11} = 6m_5^2 = 24$
 and $\hat{d} = 16$
 thus $\Delta c = 0$

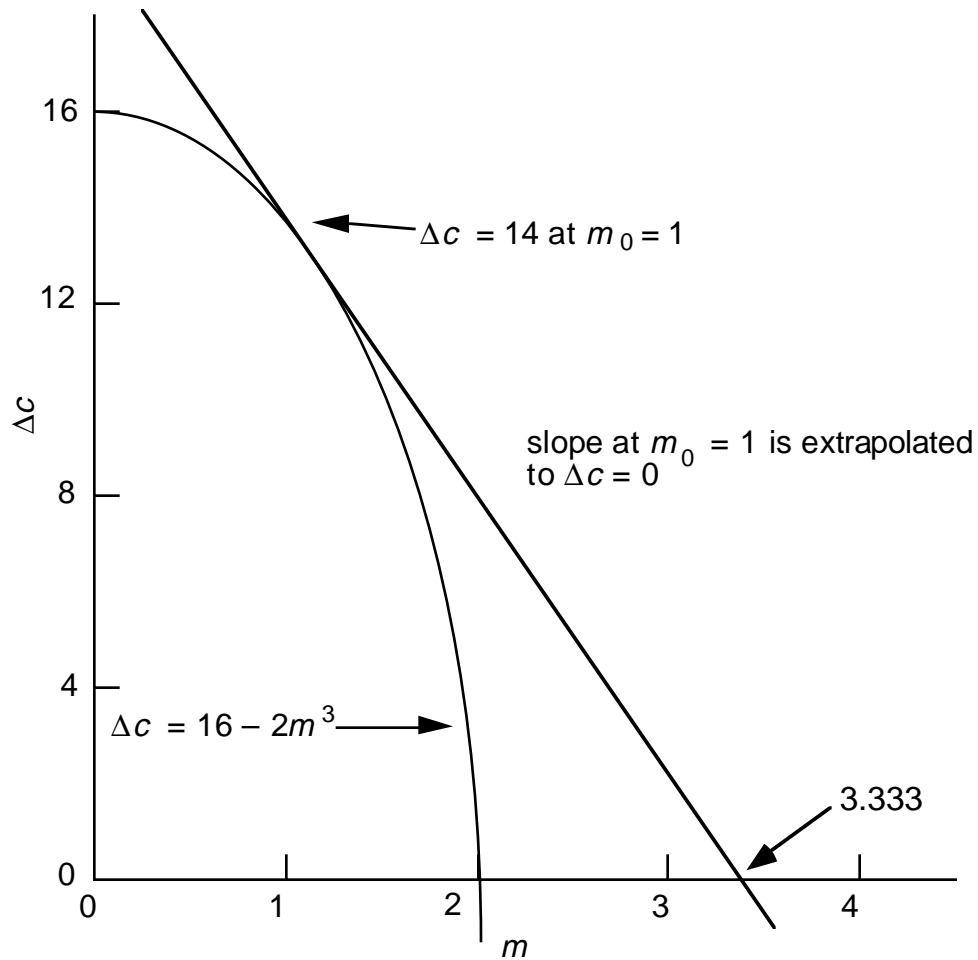
and we quit!!!! We should note that we have quit because the misfit has been reduced to some acceptable level (three significant figures in this case). The solution happens to be an integer, and we have found it to three places. Most solutions are not integers, and we must decide how many significant figures are justified. The answer depends on many things, but one of the most important is the level of noise in the data. If the data are noisy, it does not make sense to claim that a solution to seven places is meaningful.

Consider the following graph for this problem:



Note that the slope at $m = 1$, when extrapolated to give $\hat{d} = 16$, yields $m = 3.333$. The solution then iterates back down the curve to the correct solution $m = 2$.

Consider plotting Δc , rather than \hat{d} , versus m (diagram on next page). This is perhaps more useful because when we solve for Δm_i , we are always extrapolating to $\Delta c = 0$.



For this example, we see that at $m_0 = 1$, the slope $\partial g(m)/\partial m = 6$. We used this slope to extrapolate to the point where $\Delta c = 0$.

At the second iteration, $m_1 = 3.333$ and is farther from the true solution ($m_1 = 2$) than was our starting model. Also, the length squared of the misfit is 3372, much worse than the misfit (196) at our initial guess. This makes an important point: you can still get to the right answer even if some iteration takes you farther from the solution than where you have been. This is especially true for early steps in the iteration when you may not be close to the solution.

Note also that if we had started with m_0 closer to zero, the shallow slope would have sent us to an even higher value for m_1 . We would still have recovered, though (do you see why?). The shallow slope corresponds to a small singular value, illustrating the problems associated with small singular values. We will consider singular-value analysis in the next chapter.

What do you think would happen if you take $m_0 < 0$???? Would it still converge to the correct solution? Try $m_0 = -1$ if your curiosity has been piqued!

4.4.3 Nonlinear Straight-Line Example

An interesting nonlinear problem is fitting a straight line to a set of data points (y_i, z_i) which may contain errors, or noise, along *both* the y and z axes. One could cast the problem as

$$y_i = a + bz_i \quad i = 1, \dots, N \quad (4.20)$$

Assuming z were perfectly known, one obtains a solution for a, b by a linear least squares approach [see Equations (3.27) and (3.32)].

Similarly, if y were perfectly known, one obtains a solution for

$$z_i = c + dy_i \quad i = 1, \dots, N \quad (4.21)$$

again using (3.27) and (3.32). These two lines can be compared by rewriting (4.21) as a function of y , giving

$$y_i = -(c/d) + (1/d)z_i \quad i = 1, \dots, N \quad (4.22)$$

In general, $a \neq -c/d$ and $b \neq 1/d$ because in (4.20) we assumed all of the error, or misfit, was in y , while in (4.21) we assumed that all of the error was in z . Recall that the quantity being minimized in (4.20) is

$$\begin{aligned} E_1 &= [\mathbf{y}^{\text{obs}} - \mathbf{y}^{\text{pre}}]^T [\mathbf{y}^{\text{obs}} - \mathbf{y}^{\text{pre}}] \\ &= \sum_{i=1}^N (y_i - \hat{y}_i)^2 \end{aligned} \quad (4.23)$$

where \hat{y}_i is the predicted y value. The comparable quantity for (4.21) is

$$\begin{aligned} E_2 &= [\mathbf{z}^{\text{obs}} - \mathbf{z}^{\text{pre}}]^T [\mathbf{z}^{\text{obs}} - \mathbf{z}^{\text{pre}}] \\ &= \sum_{i=1}^N (z_i - \hat{z}_i)^2 \end{aligned} \quad (4.24)$$

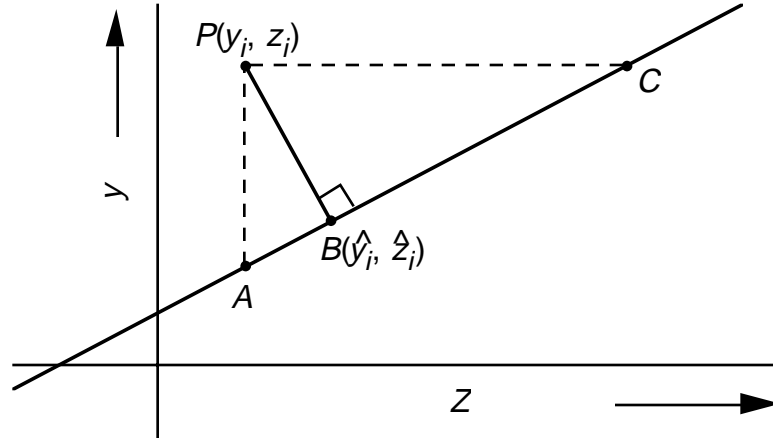
where \hat{z}_i is the predicted z value.

For the best fit line in which both y and z have errors, the function to be minimized is

$$\begin{aligned} E &= \begin{bmatrix} \mathbf{y} - \hat{\mathbf{y}} \\ \mathbf{z} - \hat{\mathbf{z}} \end{bmatrix}^T \begin{bmatrix} \mathbf{y} - \hat{\mathbf{y}} \\ \mathbf{z} - \hat{\mathbf{z}} \end{bmatrix} \\ &= \sum_{i=1}^N (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2 \end{aligned} \quad (4.25)$$

where $\mathbf{y} - \hat{\mathbf{y}}$ and $\mathbf{z} - \hat{\mathbf{z}}$ together compose a vector of dimension $2N$ if N is the number of pairs (y_i, z_i) .

Consider the following diagram:



Line PA above represents the misfit in y (4.23), line PC represents the misfit in z (4.24), while line PB represents (4.25).

In order to minimize (4.25) we must be able to write the forward problem for the predicted data (\hat{y}_i, \hat{z}_i) . Let the solution we seek be given by

$$y = m_1 + m_2 z \quad (4.26)$$

Line PB is perpendicular to (4.26), and thus has a slope of $-1/m_2$. The equation of a line through $P(\hat{y}_i, \hat{z}_i)$ with slope $-1/m_2$ is given by

$$y - y_i = -(1/m_2)(z - z_i)$$

or

$$y = (1/m_2)z_i + y_i - (1/m_2)z \quad (4.27)$$

The point $B(\hat{y}_i, \hat{z}_i)$ is thus the intersection of the lines given by (4.26) and (4.27). Equating the right-hand sides of (4.26) and (4.27) for $y = \hat{y}_i$ and $z = \hat{z}_i$ gives

$$m_1 + m_2 \hat{z}_i = (1/m_2)z_i + y_i - (1/m_2) \hat{z}_i \quad (4.28)$$

which can be solved for \hat{z}_i as

$$\hat{z}_i = \frac{-m_1 m_2 + z_i + m_2 y_i}{1 + m_2^2} \quad (4.29)$$

Rearranging (4.26) and (4.27) to give z as a function of y and again equating for $y = \hat{y}_i$ and $z = \hat{z}_i$ gives

$$(1/m_2)(\hat{y}_i - m_1) = -m_2 \hat{y}_i + z_i + m_2 y_i \quad (4.30)$$

which can be solved for \hat{y}_i as

$$\hat{y}_i = \frac{m_1 + m_2 z_i + m_2^2 y_i}{1 + m_2^2} \quad (4.31)$$

substituting (4.31) for \hat{y}_i and (4.29) for \hat{z}_i into (4.25) now gives E as a function of the unknowns m_1 and m_2 . The approach used for the linear problem was to take partials of E with respect to m_1 and m_2 , set them equal to zero, and solve for m_1 and m_2 , as was done in (3.10) and (3.11). Unfortunately, the resulting equations for partials of E with respect to m_1 and m_2 in (4.25) are not linear in m_1 and m_2 and cannot be cast in the linear form

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d} \tag{3.18}$$

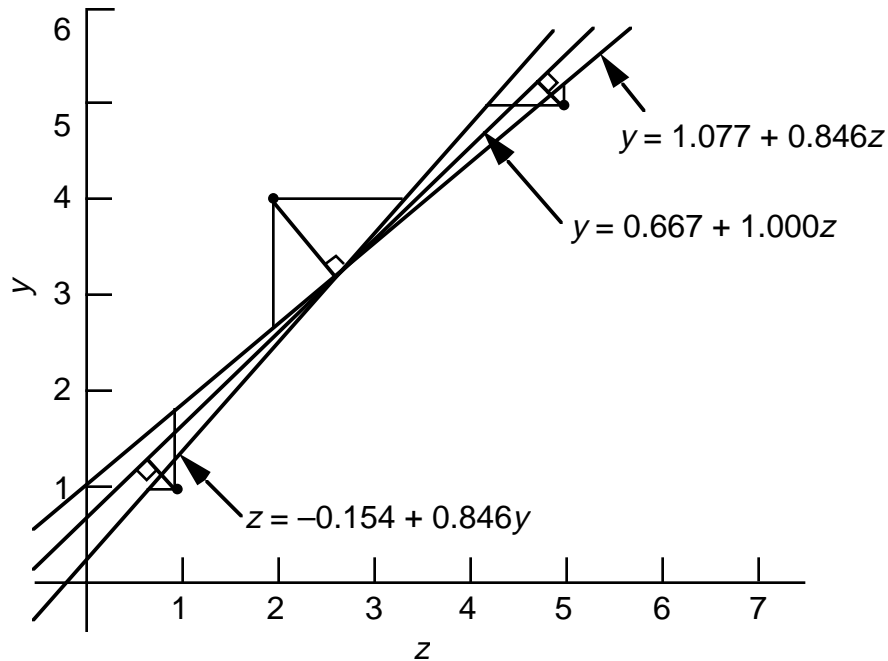
Instead, we must consider (4.29) and (4.31) to be of the form of (4.2):

$$d_i = g_i(\mathbf{m}) \tag{4.2}$$

and linearize the problem by expanding (4.29) and (4.31) in a Taylor series about some starting guesses \hat{z}_0 and \hat{y}_0 , respectively. This requires taking partials of \hat{y}_i and \hat{z}_i with respect to m_1 and m_2 , which can be obtained from (4.24) and (4.31).

Consider the following data set, shown also on the following diagram.

$y_i:$	1	4	5
$z_i:$	1	2	5



The linear least square solution to (4.20)

$$y_i = a + bz_i \quad i = 1, 2, 3 \tag{4.20}$$

is

$$y_i = 1.077 + 0.846z_i \quad i = 1, 2, 3 \tag{4.32}$$

The linear least squares solution to (4.21)

$$z_i = c + dy_i \quad i = 1, 2, 3 \quad (4.21)$$

is

$$z_i = -0.154 + 0.846y_i \quad i = 1, 2, 3 \quad (4.33)$$

For comparison with a and b above, we can rewrite (4.33) with y as a function of z as

$$y_i = 0.182 + 1.182z_i \quad i = 1, 2, 3 \quad (4.34)$$

The nonlinear least squares solution which minimizes

$$E = \sum_{i=1}^N (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2 \quad (4.25)$$

is given by

$$y_i = 0.667 + 1.000z_i \quad i = 1, 2, 3 \quad (4.35)$$

From the figure you can see that the nonlinear solution lies between the other two solutions.

It is also possible to consider a weighted nonlinear least squares best fit to a data set. In this case, we form a new E , after (3.59), as

$$E = \begin{bmatrix} \mathbf{y} - \hat{\mathbf{y}} \\ \mathbf{z} - \hat{\mathbf{z}} \end{bmatrix}^T \mathbf{W}_e \begin{bmatrix} \mathbf{y} - \hat{\mathbf{y}} \\ \mathbf{z} - \hat{\mathbf{z}} \end{bmatrix} \quad (4.36)$$

and where \mathbf{W}_e is a $2N \times 2N$ weighting matrix. The natural choice for \mathbf{W}_e is

$$\left\{ \text{cov} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \right\}^{-1}$$

the inverse data covariance matrix. If the errors in y_i, z_i are uncorrelated, then the data covariance matrix will be a diagonal matrix with the variances for y_i as the first N entries and the variances for z_i as the last N entries. If we further let V_y and V_z be the variances for y_i and z_i , respectively, then Equations (4.29) and (4.31) become

$$\hat{z}_i = \frac{-m_1 m_2 V_z + V_y z_i + m_2 V_z y_i}{m_2^2 V_z + V_y} \quad (4.37)$$

and

$$\hat{y}_i = \frac{m_1 V_y + m_2 V_y z_i + m_2^2 V_z y_i}{m_2^2 V_z + V_y} \quad (4.38)$$

If $V_z = V_y$, then dividing through either (4.37) or (4.38) by the variance returns (4.29) or (4.31). Thus we see that weighted least squares techniques are equivalent to general least squares techniques when all the data variances are equal and the errors are uncorrelated.

Furthermore, dividing both the numerator and denominator of (4.38) by V_y yields

$$\hat{y}_i = \frac{m_1 + m_2 z_i + (m_2^2 V_z y_i) / V_y}{[(m_2^2 V_z) / V_y] + 1} \quad (4.39)$$

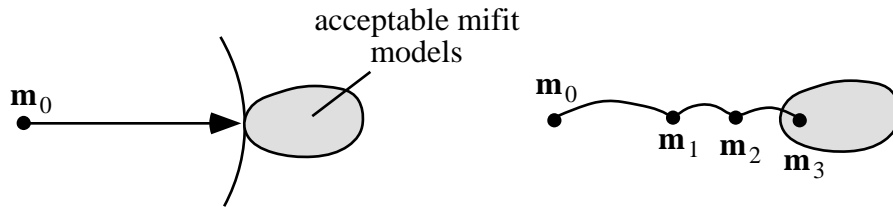
Then, in the limit that V_z goes to zero, (4.39) becomes

$$\hat{y}_i = m_1 + m_2 z_i \quad (4.40)$$

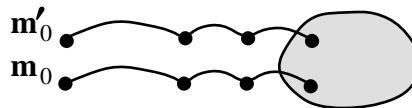
which is just the linear least squares solution of (4.20). That is, if z_i is assumed to be perfectly known ($V_z = 0$), the nonlinear problem reduces to a linear problem. Similar arguments can be made for (4.37) to show that the linear least squares solution of (4.21) results when V_y goes to zero.

4.5 Creeping vs. Jumping (Shaw and Orcutt, 1985)

The general procedure described in Section 4.3 is termed "creeping" by *Parker* (1985). It finds, from the set of acceptable misfit models, the one closest to the starting model under the Euclidian norm.



Because of the nonlinearity, often several iterations are required to reach the desired misfit. The acceptable level of misfit is free to vary and can be viewed as a parameter that controls the trade-off between satisfying the data and keeping the model perturbation small. Because the starting model itself is physically reasonable, the unphysical model estimates tend to be avoided. There are several potential disadvantages to the creeping strategy. Creeping analysis depends significantly on the choice of the initial model. If the starting model is changed slightly, a new final model may well be found. In addition, constraints applied to model perturbations may not be as meaningful as those applied directly to the model parameters.



Parker (1985) introduced an alternative approach with a simple algebraic substitution. The new method, called "jumping," directly calculates the new model in a single step rather than calculating a perturbation to the initial model. Now, any suitable norm can be applied to the model rather than to the perturbations.

This new strategy is motivated, in part, by the desire to map the neighborhood of starting models near \mathbf{m}_0 to a single final model, thus making the solution less sensitive to small change in \mathbf{m}_0 .



Let's write the original nonlinear equations as

$$\mathbf{Fm} = \mathbf{d} \tag{4.41}$$

After linearization about an initial model \mathbf{m}_0 , we have

$$\mathbf{G}\Delta\mathbf{m} = \Delta\mathbf{c} \tag{4.42}$$

when

$$\mathbf{G} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{m}} \right|_{\mathbf{m}_0}, \quad \Delta\mathbf{c} = \mathbf{d} - \mathbf{Fm}_0 \tag{4.43}$$

The algebraic substitution suggested by Parker is to simply add \mathbf{Gm}_0 to both sides, yielding

$$\mathbf{G}\Delta\mathbf{m} + \mathbf{Gm}_0 = \Delta\mathbf{c} + \mathbf{Gm}_0$$

then

$$\mathbf{G}[\Delta\mathbf{m} + \mathbf{m}_0] = \Delta\mathbf{c} + \mathbf{Gm}_0 \tag{4.44}$$

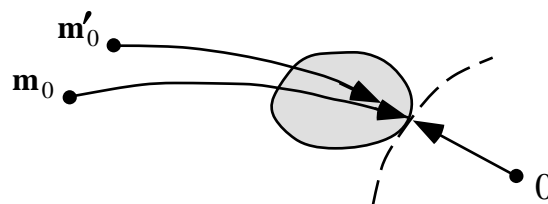
and

$$\boxed{\mathbf{Gm}_1 = \Delta\mathbf{c} + \mathbf{Gm}_0} \tag{4.45}$$

or

$$\mathbf{Gm}_1 = \mathbf{d} - \mathbf{Fm}_0 + \mathbf{Gm}_0 \tag{4.46}$$

At this point, this equation is algebraically equivalent to our starting linearized equation. But the crucial difference is that now we are solving directly for the model \mathbf{m} rather than a perturbation $\Delta\mathbf{m}$. This slight algebraic difference means we can now apply any suitable constraint to the model. A good example is a smoothing constraint. If the initial model is not smooth, applying a smoothing constraint to the model perturbations may not make sense. In the new formulation, we can apply the constraint directly to the model. In the jumping scheme, the new model is computed directly, and the norm of this model is minimized relative to an absolute origin 0 corresponding to this norm.

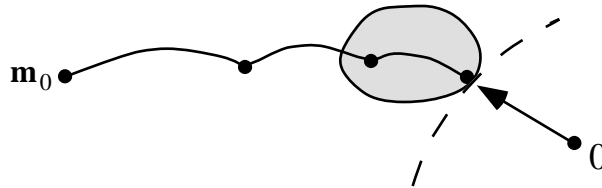


The explicit dependence on the starting model is greatly reduced.

In our example of the second derivative smoothing matrix \mathbf{D} , we can now apply this directly to the jumping equations.

$$\begin{bmatrix} \mathbf{G} \\ \theta \mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \Delta \mathbf{c} + \mathbf{G} \mathbf{m}_0 \\ \mathbf{0} \end{bmatrix} \quad (4.47)$$

We should keep in mind that since the problem is nonlinear, there is still no guarantee that the final model will be unique, and even this "jumping" scheme is iterative.



To summarize, the main advantage of the jumping scheme is that the new model is calculated directly. Thus, constraints can be imposed directly on the new model. The minimization of the squared misfit can be traded off with the constraint measures, allowing optimization of some physically significant quantity. This tends to reduce the strong dependence on the initial model that is associated with the creeping scheme.

We now turn our attention to the generalized inverse in the next three chapters. We will begin with eigenvalue problems, and then continue with singular-value problems, the generalized inverse, and ways of quantifying the quality of the solution.

CHAPTER 5: THE EIGENVALUE PROBLEM

5.1 Introduction

In Chapter 3 the emphasis was on developing inverse operators and solutions based on minimizing some combination of the (perhaps weighted) error vector \mathbf{e} and the model parameter vector \mathbf{m} . In this chapter we will change the emphasis to the operator \mathbf{G} itself. Using the concepts of vector spaces and linear transformations, we will consider how \mathbf{G} maps model parameter vectors into predicted data vectors. This approach will lead to the *generalized inverse operator*. We will see that the operators introduced in Chapter 3 can be thought of as special cases of the generalized inverse operator. The power of the generalized inverse approach, however, lies in the ability to assess the resolution and stability of the solution based on the mapping back and forth between model and data spaces.

We will begin by considering the eigenvalue problem for square matrices, and extend this to the shifted eigenvalue problem for nonsquare matrices. Once we have learned how to decompose a general matrix using the shifted eigenvalue problem, we will introduce the generalized inverse operator, show how it reduces to the operators and solutions from Chapter 3 in special cases, and consider measures of quality and stability for the solution.

The eigenvalue problem plays a central role in the vector space representation of inverse problems. Although some of this is likely to be review, it is important to cover it in some detail so that the underlying linear algebra aspects are made clear. This will lay the foundation for an understanding of the mapping of vectors back and forth between model and data spaces.

5.2 Eigenvalue Problem for the Square ($M \times M$) Matrix \mathbf{A}

5.2.1 Background

Given the following system of linear equations

$$\mathbf{Ax} = \mathbf{b} \tag{5.1}$$

where \mathbf{A} is a general $M \times M$ matrix, and \mathbf{x} and \mathbf{b} are $M \times 1$ column vectors, respectively, it is natural to immediately plunge in, calculate \mathbf{A}^{-1} , the inverse of \mathbf{A} , and solve for \mathbf{x} . Presumably, there are lots of computer programs available to invert square matrices, so why not just hand the problem over to the local computer hack and be done with it? The reasons are many, but let it suffice to say that in almost all problems in geophysics, \mathbf{A}^{-1} will not exist in the mathematical sense, and our task will be to find an approximate solution, we hope along with some measure of the quality of that solution.

One approach to solving Equation (5.1) above is through the eigenvalue problem for \mathbf{A} . It will have the benefit that, in addition to finding \mathbf{A}^{-1} when it exists, it will lay the groundwork for finding approximate solutions for the vast majority of situations where the exact, mathematical inverse does not exist.

In the eigenvalue problem, the matrix \mathbf{A} is thought of as a linear transformation that transforms one M -dimensional vector into another M -dimensional vector. Eventually, we will seek the particular \mathbf{x} that solves $\mathbf{A}\mathbf{x} = \mathbf{b}$, but the starting point is the more general problem of transforming any particular M -dimensional vector into another M -dimensional vector.

We begin by defining M space as the space of all M -dimensional vectors. For example, 2 space is the space of all vectors in a plane. The eigenvalue problem can then be defined as follows:

For \mathbf{A} , an $M \times M$ matrix, find all vectors \mathbf{s} in M space such that

$$\mathbf{A}\mathbf{s} = \lambda \mathbf{s} \tag{5.2}$$

where λ is a constant called the eigenvalue and \mathbf{s} is the associated eigenvector.

In words, this means find all vectors \mathbf{s} that, when operated on by \mathbf{A} , return a vector that points in the same direction (up to the sign of the vector) with a length scaled by the constant λ .

5.2.2 How Many Eigenvalues, Eigenvectors?

If, $\mathbf{A}\mathbf{s} = \lambda\mathbf{s}$, then

$$[\mathbf{A} - \lambda\mathbf{I}_M]\mathbf{s} = \mathbf{0}_M \tag{5.3}$$

where $\mathbf{0}_M$ is an $M \times 1$ vector of zeros. Equation (5.3) is called a *homogeneous equation* because the right-hand side consists of zeros. It has a nontrivial solution (e.g., $\mathbf{s} \neq [0, 0, \dots, 0]^T$) if and only if

$$|\mathbf{A} - \lambda\mathbf{I}_M| = 0 \tag{5.4}$$

where

$$|\mathbf{A} - \lambda\mathbf{I}_M| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} - \lambda & & a_{2M} \\ \vdots & & \ddots & \vdots \\ a_{M1} & \cdots & & a_{MM} - \lambda \end{vmatrix} \tag{5.5}$$

Equation (5.4), also called the characteristic equation, is a polynomial of order M in λ of the form

$$\lambda^M + C_{M-1}\lambda^{M-1} + C_{M-2}\lambda^{M-2} + \cdots + C_0\lambda^0 = 0 \tag{5.6}$$

Equation (5.6) has M and only M roots for λ . In general, these roots may be complex (shudder!). However, if \mathbf{A} is Hermitian, then all of the M roots are real (whew!). They may, however, have repeated values (ugh!), be negative (ouch), or zero (yuk).

We may write the M roots for λ as

$$\lambda_1, \lambda_2, \dots, \lambda_M$$

For each λ_i an associated eigenvector \mathbf{s}_i which solves Equation (5.2) can be (easily) found. It is important to realize that the ordering of the λ_i is completely arbitrary.

Equation (5.2), then, holds for M values of λ_i . That is, we have

$$\begin{aligned} \lambda = \lambda_1, \mathbf{s} &= (s_1^1, s_2^1, \dots, s_M^1)^T = \mathbf{s}_1 \\ \lambda = \lambda_2, \mathbf{s} &= (s_1^2, s_2^2, \dots, s_M^2)^T = \mathbf{s}_2 \\ &\vdots \\ \lambda = \lambda_M, \mathbf{s} &= (s_1^M, s_2^M, \dots, s_M^M)^T = \mathbf{s}_M \end{aligned} \tag{5.7}$$

About the above notation: s_i^j is the i th component of the j th eigenvector. It turns out to be very inconvenient to use superscripts to denote which eigenvector it is, so \mathbf{s}_j will denote the j th eigenvector. Thus, if I use *only* a subscript, it refers to the number of the eigenvector. Of course, if it is a vector instead of a component of a vector, it should always appear as a bold-face letter.

The length of eigenvectors is arbitrary. To see this note that if

$$\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \tag{5.8a}$$

then

$$\mathbf{A}(2\mathbf{s}_i) = \lambda_i(2\mathbf{s}_i) \tag{5.8b}$$

since \mathbf{A} is a linear operator.

5.2.3 The Eigenvalue Problem in Matrix Notation

Now that we know that there are M values of λ_i and M associated eigenvectors, we can write Equation (5.2) as

$$\boxed{\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \quad i = 1, 2, \dots, M} \tag{5.9}$$

Consider, then, the following matrices:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \lambda_M \end{bmatrix} \tag{5.10}$$

$M \times M$

and

$$\mathbf{S} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.11)$$

$M \times M$

where the i th column of \mathbf{S} is the eigenvector \mathbf{s}_i associated with the i th eigenvalue λ_i . Then Equation (5.9) can be written in compact, matrix notation as

$$\mathbf{AS} = \mathbf{S}\Lambda \quad (5.12)$$

where the order of the matrices on the right-hand side is important! To see this, let us consider the i th columns of $[\mathbf{AS}]$ and $[\mathbf{S}\Lambda]$, respectively. The first component of the i th column of $[\mathbf{AS}]$ is given by

$$(\mathbf{AS})_{li} = \sum_{k=1}^M a_{lk} s_{ki} = \sum_{k=1}^M a_{lk} s_k^i \quad (5.13)$$

where $s_{ki} = s_k^i$ is the k th component of the i th eigenvector.

Similarly, components 2 through M of the i th column are given by

$$\begin{aligned} (\mathbf{AS})_{2i} &= \sum_{k=1}^M a_{2k} s_{ki} = \sum_{k=1}^M a_{2k} s_k^i \\ &\vdots \\ (\mathbf{AS})_{Mi} &= \sum_{k=1}^M a_{Mk} s_{ki} = \sum_{k=1}^M a_{Mk} s_k^i \end{aligned} \quad (5.14)$$

Therefore, the i th column of $[\mathbf{AS}]$ is given by

$$[\mathbf{AS}]_i = \begin{bmatrix} \sum_{k=1}^M a_{1k} s_k^i \\ \sum_{k=1}^M a_{2k} s_k^i \\ \vdots \\ \sum_{k=1}^M a_{Mk} s_k^i \end{bmatrix} = \mathbf{A}\mathbf{s}_i \quad (5.15)$$

That is, the i th column of $[\mathbf{AS}]$ is given by the product of \mathbf{A} and the i th eigenvector \mathbf{s}_i .

Now, the first element in the i th column of $[\mathbf{S}\Lambda]$ is found as follows:

$$[\mathbf{S}\Lambda]_{li} = \sum_{k=1}^M s_{lk} \Lambda_{ki} = \sum_{k=1}^M s_{lk} \Lambda_{ki} \quad (5.16)$$

But,

$$\Lambda_{ki} = \delta_{ki} \lambda_i = \begin{cases} 0, & k \neq i \\ \lambda_i, & k = i \end{cases} \quad (5.17)$$

where δ_{ki} is the Kronecker delta. Therefore,

$$[\mathbf{S}\Lambda]_{1i} = s_1^i \lambda_i \quad (5.18)$$

Entries 2 through M are then given by

$$\begin{aligned} [\mathbf{S}\Lambda]_{2i} &= \sum_{k=1}^M s_2^k \Lambda_{ki} = s_2^i \lambda_i \\ &\vdots \\ [\mathbf{S}\Lambda]_{Mi} &= \sum_{k=1}^M s_M^k \Lambda_{ki} = s_M^i \lambda_i \end{aligned} \quad (5.19)$$

Thus, the i th column of $[\mathbf{S}\Lambda]$ is given by

$$\begin{bmatrix} s_1^i \\ s_2^i \\ \vdots \\ s_M^i \end{bmatrix} = \lambda_i \begin{bmatrix} s_1^i \\ s_2^i \\ \vdots \\ s_M^i \end{bmatrix} = \lambda_i \mathbf{s}_i \quad (5.20)$$

Thus, we have that the original equation defining the eigenvalue problem [Equation (5.9)]

$$\mathbf{A}\mathbf{s}_i = \lambda_i \mathbf{s}_i$$

is given by the i th columns of the matrix equation

$$\mathbf{A}\mathbf{S} = \mathbf{S}\Lambda \quad (5.12)$$

Consider, on the other hand, the i th column of $\Lambda\mathbf{S}$

$$\begin{aligned} [\Lambda\mathbf{S}]_{1i} &= \sum_{k=1}^M \Lambda_{1k} s_{ki} = \lambda_1 s_1^i \\ [\Lambda\mathbf{S}]_{2i} &= \sum_{k=1}^M \Lambda_{2k} s_{ki} = \lambda_2 s_2^i \\ &\vdots \end{aligned}$$

$$[\Lambda \mathbf{S}]_{Mi} = \sum_{k=1}^M \Lambda_{Mk} s_{ki} = \lambda_{1M} s_M^i \quad (5.21a)$$

Therefore, the i th column of $[\Lambda \mathbf{S}]$ is given by

$$[\Lambda \mathbf{S}]_i = \Lambda \mathbf{s}_i \quad (5.21b)$$

That is, the product of Λ with the i th eigenvector \mathbf{s}_i . Clearly, this is not equal to $\Lambda \mathbf{s}_i$ above in Equation (5.9).

5.2.4 Summarizing the Eigenvalue Problem for \mathbf{A}

In review, we found that we could write the eigenvalue problem for $\mathbf{A}\mathbf{x} = \mathbf{b}$ as

$$\mathbf{A}\mathbf{S} = \mathbf{S}\Lambda \quad (5.12)$$

where

$$\mathbf{S} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.11)$$

is an $M \times M$ matrix, each column of which is an eigenvector \mathbf{s}_i such that

$$\mathbf{A}\mathbf{s}_i = \lambda_i \mathbf{s}_i \quad (5.9)$$

and where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_M \end{bmatrix} \quad (5.10)$$

is a diagonal $M \times M$ matrix with the eigenvalue λ_i of $\mathbf{A}\mathbf{s}_i = \lambda_i \mathbf{s}_i$ along the diagonal and zeros everywhere else.

5.3 Geometrical Interpretation of the Eigenvalue Problem for Symmetric \mathbf{A}

5.3.1 Introduction

It is possible, of course, to cover the mechanics of the eigenvalue problem without ever considering a geometrical interpretation. There is, however, a wealth of information to be learned

from looking at the geometry associated with the operator \mathbf{A} . This material is not covered in Menke's book, although some of it is covered in *Statistics and Data Analysis in Geology*, 2nd Edition, 1986, pages 131–139, by J. C. Davis, published by John Wiley and Sons.

We take as an example the symmetric 2×2 matrix \mathbf{A} given by

$$\mathbf{A} = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \quad (5.22)$$

Working with a symmetric matrix assures real eigenvalues. We will discuss the case for the general square matrix later.

The eigenvalue problem is solved by

$$|\mathbf{A} - \lambda\mathbf{I}| = 0 \quad (5.23)$$

where it is found that $\lambda_1 = 7$, $\lambda_2 = 2$, and the associated eigenvectors are given by

$$\mathbf{s}_1 = [0.894, 0.447]^T \quad (5.24a)$$

and

$$\mathbf{s}_2 = [-0.447, 0.894]^T \quad (5.24b)$$

respectively. Because \mathbf{A} is symmetric, \mathbf{s}_1 and \mathbf{s}_2 are perpendicular to each other. The length of eigenvectors is arbitrary, but for orthogonal eigenvectors it is common to normalize them to unit length, as done in Equations (5.24a) and (5.24b). The sign of eigenvectors is arbitrary as well, and I have chosen the signs for \mathbf{s}_1 and \mathbf{s}_2 in Equations (5.24a) and (5.24b) for convenience when I later relate them to orthogonal transformations.

5.3.2 Geometrical Interpretation

The system of linear equations

$$\mathbf{Ax} = \mathbf{b} \quad (5.1)$$

implies that the $M \times M$ matrix \mathbf{A} transforms $M \times 1$ vectors \mathbf{x} into $M \times 1$ vectors \mathbf{b} . In our example, $M = 2$, and hence $\mathbf{x} = [x_1, x_2]^T$ represents a point in 2-space, and $\mathbf{b} = [b_1, b_2]^T$ is just another point in the same plane.

Consider the unit circle given by

$$\mathbf{x}^T\mathbf{x} = x_1^2 + x_2^2 = 1 \quad (5.25)$$

Matrix \mathbf{A} maps every point on this circle onto another point. The eigenvectors \mathbf{s}_1 and \mathbf{s}_2 , having unit length, are points on this circle. Then, since

$$\mathbf{As}_i = \lambda_i\mathbf{s}_i \quad (5.8a)$$

we have

$$\mathbf{A}\mathbf{s}_1 = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix} = \begin{bmatrix} 6.258 \\ 3.129 \end{bmatrix} = 7 \begin{bmatrix} 0.894 \\ 0.447 \end{bmatrix} \quad (5.26a)$$

and

$$\mathbf{A}\mathbf{s}_2 = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} -0.447 \\ 0.894 \end{bmatrix} = \begin{bmatrix} -0.894 \\ 1.788 \end{bmatrix} = 2 \begin{bmatrix} -0.447 \\ 0.894 \end{bmatrix} \quad (5.26b)$$

As expected, when \mathbf{A} operates on an eigenvector, it returns another vector parallel (or antiparallel) to the eigenvector, scaled by the eigenvalue. When \mathbf{A} operates on any direction different from the eigenvector directions, it returns a vector that is not parallel (or antiparallel) to the original vector.

What is the shape mapped out by \mathbf{A} operating on the unit circle? We have already seen where \mathbf{s}_1 and \mathbf{s}_2 map. If we map out this transformation for the unit circle, we get an ellipse, and this is an important element of the geometrical interpretation. What is the equation for this ellipse? We begin with our unit circle given by

$$\mathbf{x}^T\mathbf{x} = 1 \quad (5.25)$$

and

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (5.1)$$

then

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (5.27)$$

assuming \mathbf{A}^{-1} exists. Then, substituting (5.27) into (5.25) we get for a general \mathbf{A}

$$[\mathbf{A}^{-1}\mathbf{b}]^T[\mathbf{A}^{-1}\mathbf{b}] = \mathbf{b}^T[\mathbf{A}^{-1}]^T\mathbf{A}^{-1}\mathbf{b} = 1 \quad (5.28a)$$

Where \mathbf{A} is symmetric,

$$[\mathbf{A}^{-1}]^T = \mathbf{A}^{-1} \quad (5.28b)$$

and in this case

$$\mathbf{b}^T\mathbf{A}^{-1}\mathbf{A}^{-1}\mathbf{b} = 1 \quad (5.28c)$$

After some manipulations, Equation (5.28) for the present choice of \mathbf{A} given in Equation (5.22) gives

$$\frac{b_1^2}{15.077} + \frac{b_1b_2}{5.444} + \frac{b_2^2}{4.900} = 1 \quad (5.29)$$

which we recognize as the equation of an ellipse inclined to the b_1 and b_2 axes.

We can now make the following geometrical interpretation on the basis of the eigenvalue problem:

1. *The major and minor axis directions are given by the directions of the eigenvectors \mathbf{s}_1 and \mathbf{s}_2 .*
2. *The lengths of the semimajor and semiminor axes of the ellipse are given by the absolute values of eigenvalues λ_1 and λ_2 .*
3. *The columns of \mathbf{A} (i.e., $[6, 2]^T$ and $[2, 3]^T$ in our example) are vectors from the origin to points on the ellipse.*

The third observation follows from the fact that \mathbf{A} operating on $[1, 0]^T$ and $[0, 1]^T$ return the first and second columns of \mathbf{A} , respectively, and both unit vectors clearly fall on the unit circle.

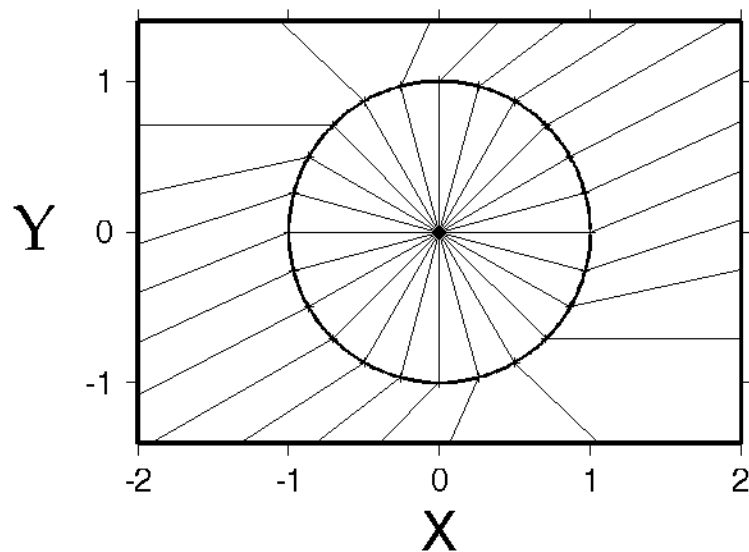
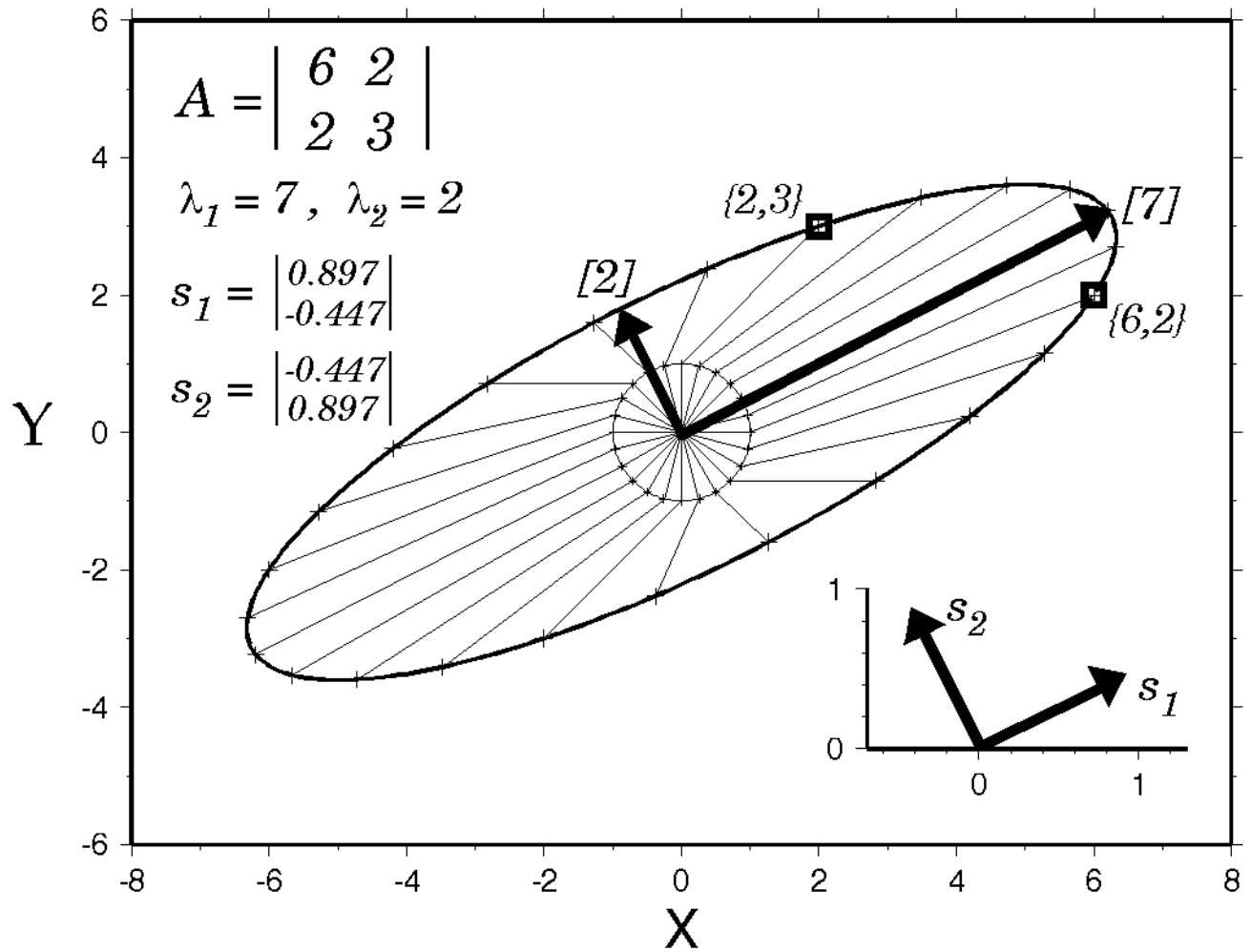
If one of the eigenvalues is negative, the unit circle is still mapped onto an ellipse, but some points around the unit circle will be mapped back through the circle to the other side. The absolute value of the eigenvalue still gives the length of the semiaxis.

The geometrical interpretation changes somewhat if \mathbf{A} is not symmetrical. The unit circle is still mapped onto an ellipse, and the columns of \mathbf{A} still represent vectors from the origin to points on the ellipse. Furthermore, the absolute values of the eigenvalues still give the lengths of the semimajor and semiminor axes. The only difference is that the eigenvectors \mathbf{s}_1 and \mathbf{s}_2 will no longer be orthogonal to each other and will not give the directions of the semimajor and semiminor axes.

The geometrical interpretation holds for larger-dimensional matrices. In the present 2×2 case for \mathbf{A} , the unit circle maps onto an ellipse. For a 3×3 case, a unit sphere maps onto an ellipsoid. In general, the surface defined by the eigenvalue problem is of dimension one less than the dimension of \mathbf{A} .

The following page presents a diagram of this geometry.

Geometry of the Eigenvalue Problem



*Blowup of
Central Portion
from Above*

5.3.3 Coordinate System Rotation

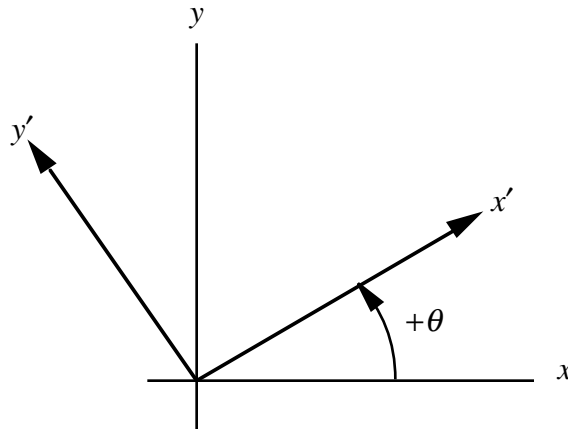
For the case of a symmetric \mathbf{A} matrix, it is possible to relate the eigenvectors of \mathbf{A} with a coordinate transformation that diagonalizes \mathbf{A} . For the example given in Equation (5.22), we construct the eigenvector matrix \mathbf{S} with the eigenvectors \mathbf{s}_1 and \mathbf{s}_2 as the first and second columns, respectively. Then

$$\mathbf{S} = \begin{bmatrix} 0.894 & -0.447 \\ 0.447 & 0.894 \end{bmatrix} \quad (5.30)$$

Since \mathbf{s}_1 and \mathbf{s}_2 are unit vectors perpendicular to one another, we see that \mathbf{S} represents an orthogonal transformation, and

$$\mathbf{S}\mathbf{S}^T = \mathbf{S}^T\mathbf{S} = \mathbf{I}_2 \quad (5.31)$$

Now consider the coordinate transformation shown below



where

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.32a)$$

and

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T}^T \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (5.32b)$$

where \mathbf{T} transforms $[x, y]^T$ to $[x', y']^T$ and \mathbf{T}^T transforms $[x', y']^T$ to $[x, y]$, where θ is positive for counterclockwise rotation from x to x' , and where

$$\mathbf{T}^T\mathbf{T} = \mathbf{T}\mathbf{T}^T = \mathbf{I}_2 \quad (5.32c)$$

It is always possible to choose the signs of \mathbf{s}_1 and \mathbf{s}_2 such that they can be associated with x' and y' , as I have done in Equations (5.24a) and (5.24b). Looking at the components of \mathbf{s}_1 , then, we see that $\theta = 26.6^\circ$, and we further note that

$$\mathbf{S} = \mathbf{T}^T \quad (5.33)$$

The matrix \mathbf{A} can also be thought of as a symmetric second-order tensor (such as stress or strain, for example) in the original (x, y) coordinate system.

Tensors can also be rotated into the (x', y') coordinate system as

$$\mathbf{A}' = \mathbf{TAT}^T \quad (5.34a)$$

where \mathbf{A}' is the rotated tensor. Using Equation (5.33) to replace \mathbf{T} in Equation (5.34) yields

$$\mathbf{A}' = \mathbf{S}^T \mathbf{A} \mathbf{S} \quad (5.34b)$$

If you actually perform this operation on the example, you will find that \mathbf{A}' is given by

$$\mathbf{A}' = \begin{bmatrix} 7 & 0 \\ 0 & 2 \end{bmatrix} \quad (5.35)$$

Thus, we find that in the new coordinate system defined by the \mathbf{s}_1 and \mathbf{s}_2 axes, \mathbf{A}' is a diagonal matrix with the diagonals given by the eigenvalues of \mathbf{A} . If we were to write the equation for the ellipse in Equation (5.28) in the (x', y') coordinates, we would find

$$\frac{(x')^2}{49} + \frac{(y')^2}{4} = 1 \quad (5.36)$$

which is just the equation of an ellipse with semimajor and semiminor axes aligned with the coordinate axes and of length 7 and 2, respectively. This new (x', y') coordinate system is often called the principal coordinate system.

In summary, we see that the eigenvalue problem for symmetric \mathbf{A} results in an ellipse whose semimajor and semiminor axis directions and lengths are given by the eigenvectors and eigenvalues of \mathbf{A} , respectively, and that these eigenvectors can be thought of as the orthogonal transformation that rotates \mathbf{A} into a principal coordinate system where the ellipse is aligned with the coordinate axes.

5.3.4 Summarizing Points

A few points can be made:

1. The trace of a matrix is unchanged by orthogonal transformations, where the trace is defined as the sum of the diagonal terms, that is

$$\text{trace}(\mathbf{A}) = \sum_{i=1}^M a_{ii} \quad (2.12)$$

This implies that $\text{trace}(\mathbf{A}) = \text{trace}(\mathbf{A}')$. You can use this fact to verify that you have correctly found the eigenvalues.

2. If the eigenvalues had been repeated, it would imply that the length of the two axes of the ellipse are the same. That is, the ellipse would degenerate into a circle. In this case, the uniqueness of the directions of the eigenvectors vanishes. Any two vectors, preferably orthogonal, would suffice.
3. If one of the eigenvalues is zero, it means the minor axis has zero length, and the ellipse collapses into a straight line. No information about the direction perpendicular to the major axis can be obtained.

5.4 Decomposition Theorem for Square \mathbf{A}

We have considered the eigenvalue problem for \mathbf{A} . Now it is time to turn our attention to the eigenvalue problem for \mathbf{A}^T . It is important to do so because we will learn that \mathbf{A}^T has the same eigenvalues as \mathbf{A} , but in general, different eigenvectors. We will be able to use the information about shared eigenvalues to decompose \mathbf{A} into a product of matrices.

5.4.1 The Eigenvalue Problem for \mathbf{A}^T

The eigenvalue problem for \mathbf{A}^T is given by

$$\mathbf{A}^T \mathbf{r}_i = \eta_i \mathbf{r}_i \quad (5.37)$$

where η_i is the eigenvalue and \mathbf{r}_i is the associated $M \times 1$ column eigenvector. Proceeding in a manner similar to the eigenvalue problem for \mathbf{A} , we have

$$[\mathbf{A}^T - \eta \mathbf{I}_M] \mathbf{r} = \mathbf{0}_M \quad (5.38)$$

This has nontrivial solutions for \mathbf{r} if and only if

$$|\mathbf{A}^T - \eta \mathbf{I}_M| \equiv 0 \quad (5.39)$$

But mathematically, $|\mathbf{B}| = |\mathbf{B}^T|$. That is, the determinant is unchanged when you interchange rows and columns! The implication is that the M th-order polynomial in η

$$\eta^M + b_{M-1} \eta^{M-1} + \dots + b_0 \eta^0 = 0 \quad (5.40)$$

is exactly the same as the M th-order polynomial in λ for the eigenvalue problem for \mathbf{A} . That is, \mathbf{A} and \mathbf{A}^T have exactly the same eigenvalues. Therefore,

$$\eta_i = \lambda_i \quad (5.41)$$

5.4.2 Eigenvectors for \mathbf{A}^T

In general, the eigenvectors \mathbf{s} for \mathbf{A} and the eigenvectors \mathbf{r} for \mathbf{A}^T are not the same. Nevertheless, we can write the eigenvalue problem for \mathbf{A}^T in matrix notation as

$$\mathbf{A}^T \mathbf{R} = \mathbf{R} \Lambda \quad (5.42)$$

where

$$\mathbf{R} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.43)$$

is an $M \times M$ matrix of the eigenvectors \mathbf{r}_i of $\mathbf{A}^T \mathbf{r}_i = \lambda_i \mathbf{r}_i$, and where

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_M \end{bmatrix} \quad (5.44)$$

is the same eigenvalue matrix shared by \mathbf{A} .

5.4.3 Decomposition Theorem for Square Matrices

Statement of the Theorem

We are now in a position to decompose the square matrix \mathbf{A} as the product of three matrices.

Theorem: The square $M \times M$ matrix \mathbf{A} can be written as

$$\mathbf{A} = \mathbf{S} \Lambda \mathbf{R}^T \quad (5.45)$$

where \mathbf{S} is the $M \times M$ matrix whose columns are the eigenvectors of \mathbf{A} , \mathbf{R} is the $M \times M$ matrix whose columns are the eigenvectors of \mathbf{A}^T , and Λ is the $M \times M$ diagonal matrix whose diagonal entries are the eigenvalues shared by \mathbf{A} and \mathbf{A}^T and whose off-diagonal entries are zero.

We will use this theorem to find the inverse of \mathbf{A} , if it exists. In this section, then, we will go through the steps to show that the decomposition theorem is true. This involves combining the results for the eigenvalue problems for \mathbf{A} and \mathbf{A}^T .

Proof of the Theorem

Step 1. Combining the results for \mathbf{A} and \mathbf{A}^T . We start with Equation (5.12):

$$\mathbf{A} \mathbf{S} = \mathbf{S} \Lambda \quad (5.12)$$

Premultiply Equation (5.12) by \mathbf{R}^T , which gives

$$\mathbf{R}^T \mathbf{A} \mathbf{S} = \mathbf{R}^T \mathbf{S} \mathbf{\Lambda} \quad (5.46)$$

Now, returning to Equation (5.42)

$$\mathbf{A}^T \mathbf{R} = \mathbf{R} \mathbf{\Lambda} \quad (5.42)$$

Taking the transpose of Equation (5.42)

$$[\mathbf{A}^T \mathbf{R}]^T = (\mathbf{R} \mathbf{\Lambda})^T \quad (5.47a)$$

or

$$\mathbf{R}^T [\mathbf{A}^T]^T = \mathbf{\Lambda}^T \mathbf{R}^T \quad (5.47b)$$

or

$$\mathbf{R}^T \mathbf{A} = \mathbf{\Lambda} \mathbf{R}^T \quad (5.47c)$$

since

$$[\mathbf{A}^T]^T = \mathbf{A}$$

and

$$\mathbf{\Lambda}^T = \mathbf{\Lambda}$$

Next, we postmultiply Equation (5.47c) by \mathbf{S} to get

$$\mathbf{R}^T \mathbf{A} \mathbf{S} = \mathbf{\Lambda} \mathbf{R}^T \mathbf{S} \quad (5.48)$$

Noting that Equations (5.46) and (5.48) have the same left-hand sides, we have

$$\mathbf{R}^T \mathbf{S} \mathbf{\Lambda} = \mathbf{\Lambda} \mathbf{R}^T \mathbf{S} \quad (5.49)$$

or

$$\mathbf{R}^T \mathbf{S} \mathbf{\Lambda} - \mathbf{\Lambda} \mathbf{R}^T \mathbf{S} = \mathbf{0}_M \quad (5.50)$$

where $\mathbf{0}_M$ is an $M \times M$ matrix of all zeroes.

Step 2. Showing that $\mathbf{R}^T \mathbf{S} = \mathbf{I}_M$. In order to proceed beyond Equation (5.50), we need to show that

$$\mathbf{R}^T \mathbf{S} = \mathbf{I}_M \quad (5.51)$$

This means two things. First, it means that

$$[\mathbf{R}^T]^{-1} = \mathbf{S} \quad (5.52)$$

and

$$\mathbf{S}^{-1} = \mathbf{R}^T \quad (5.53)$$

Second, it means the dot product of the eigenvectors in \mathbf{R} (the columns of \mathbf{R}) with the eigenvectors in \mathbf{S} is given by

$$\mathbf{r}_i^T \mathbf{s}_j = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (5.54)$$

That is, if $i \neq j$, \mathbf{r}_i has no projection onto \mathbf{s}_j . If $i = j$, then the projection of \mathbf{r}_i onto \mathbf{s}_i can be made to be 1. I say that it can be made to be 1 because the lengths of eigenvectors is arbitrary. Thus, if the two vectors have a nonzero projection onto each other, the length of one, or the other, or some combination of both vectors, can be changed such that the projection onto each other is 1.

Let us consider, for the moment, the matrix product $\mathbf{R}^T \mathbf{S}$, and let

$$\mathbf{W} = \mathbf{R}^T \mathbf{S} \quad (5.55)$$

Then Equation (5.50) implies

$$\begin{bmatrix} (\lambda_1 - \lambda_1)W_{11} & (\lambda_2 - \lambda_1)W_{12} & \cdots & (\lambda_M - \lambda_1)W_{1M} \\ (\lambda_1 - \lambda_2)W_{21} & (\lambda_2 - \lambda_2)W_{22} & \cdots & (\lambda_M - \lambda_2)W_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_1 - \lambda_M)W_{M1} & (\lambda_2 - \lambda_M)W_{M2} & \cdots & (\lambda_M - \lambda_M)W_{MM} \end{bmatrix} = \mathbf{0}_M \quad (5.56)$$

Thus, independent of the values for W_{ii} , the diagonal entries, we have that:

$$\begin{bmatrix} 0 & (\lambda_2 - \lambda_1)W_{12} & \cdots & (\lambda_M - \lambda_1)W_{1M} \\ (\lambda_1 - \lambda_2)W_{21} & 0 & \cdots & (\lambda_M - \lambda_2)W_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_1 - \lambda_M)W_{M1} & (\lambda_2 - \lambda_M)W_{M2} & \cdots & 0 \end{bmatrix} = \mathbf{0}_M \quad (5.57)$$

If none of the eigenvalues is repeated (i.e., $\lambda_i \neq \lambda_j$), then

$$(\lambda_i - \lambda_j) \neq 0 \quad (5.58)$$

and it follows that

$$W_{ij} = 0 \quad i \neq j \quad (5.59)$$

If $\lambda_i = \lambda_j$ for some pair of eigenvalues, then it can still be shown that Equation (5.59) holds. The explanation rests on the fact that when an eigenvalue is repeated, there is a plane (or hyper-plane if $M > 3$) associated with the eigenvalues rather than a single direction, as is the case for the eigenvector associated with a nonrepeated eigenvalue. One

has the freedom to choose the eigenvectors \mathbf{r}_i , \mathbf{s}_i and \mathbf{r}_j , \mathbf{s}_j in such a way that $W_{ij} = 0$, while still having the eigenvectors span the appropriate planes. Needless to say, the proof is much more complicated than for the case without repeated eigenvalues, and will be left to the student as an exercise.

The end result, however, is that we are left with

$$\mathbf{R}^T \mathbf{S} = \begin{bmatrix} W_{11} & 0 & \cdots & 0 \\ 0 & W_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_{MM} \end{bmatrix} \quad (5.60)$$

We recognize the W_{ii} entries as the dot product of \mathbf{r}_i and \mathbf{s}_i , given by

$$W_{ii} = \sum_{k=1}^M r_{ki} s_{ki} = \sum_{k=1}^M r_k^i s_k^i = \mathbf{r}_i^T \mathbf{s}_i \quad (5.61)$$

If $W_{ii} \neq 0$, then we can make $W_{ii} = 1$ by scaling \mathbf{r}_i , \mathbf{s}_i , or some combination of both. We can claim that $W_{ii} \neq 0$ as follows:

1. \mathbf{r}_i is orthogonal to \mathbf{s}_j , $i \neq j$.

That is, \mathbf{r}_i , a vector in M space, is perpendicular to $M - 1$ other vectors in M space. These $M - 1$ vectors are not all perpendicular to each other, or our work would be done.

2. However, the vectors in \mathbf{R} (and \mathbf{S}) span M space.

That is, one can write an arbitrary vector in M space as a linear combination of the vectors in \mathbf{R} (or \mathbf{S}). Thus, \mathbf{r}_i , which has no projection on $M - 1$ independent vectors in M space, must have some projection on the only vector left in \mathbf{S} , \mathbf{s}_i .

Since the projection is nonzero, one has the freedom to choose the \mathbf{r}_i and \mathbf{s}_j such that

$$W_{ii} = \sum_{k=1}^M r_{ki} s_{ki} = \sum_{k=1}^M r_k^i s_k^i = \mathbf{r}_i^T \mathbf{s}_i = 1 \quad (5.62)$$

Thus, finally, we have shown that it is possible to scale the vectors in \mathbf{R} and/or \mathbf{S} such that

$$\mathbf{R}^T \mathbf{S} = \mathbf{I}_M \quad (5.63)$$

This means that

$$\mathbf{R}^T = \mathbf{S}^{-1} \quad (5.64)$$

and

$$[\mathbf{R}^T \mathbf{S}]^T = \mathbf{I}^T = \mathbf{I} = \mathbf{S}^T \mathbf{R} \quad (5.65)$$

and

$$\mathbf{S}^T = \mathbf{R}^{-1} \quad (5.66)$$

Thus, the inverse of one is the transpose of the other, etc.

Before leaving this subject, I should emphasize that \mathbf{R} and \mathbf{S} are not orthogonal matrices. This is because, in general,

$$\mathbf{R}^T \mathbf{R} \neq \mathbf{I}_M \neq \mathbf{R} \mathbf{R}^T \quad (5.67a)$$

and

$$\mathbf{S}^T \mathbf{S} \neq \mathbf{I}_M \neq \mathbf{S} \mathbf{S}^T \quad (5.67b)$$

We cannot even say that \mathbf{S} and \mathbf{R} are orthogonal to each other. It is true that \mathbf{r}_i is perpendicular to \mathbf{s}_j , $i \neq j$, because we have shown that

$$\mathbf{r}_i^T \mathbf{s}_j = 0 \quad i \neq j$$

but

$$\mathbf{r}_i^T \mathbf{s}_i = 1 \quad (5.68)$$

does not imply that \mathbf{r}_i is parallel to \mathbf{s}_i . Recall that

$$\mathbf{r}_i^T \mathbf{s}_i = \|\mathbf{r}_i\| \|\mathbf{s}_i\| \cos \theta \quad (5.69)$$

where θ is the angle between \mathbf{r}_i and \mathbf{s}_i . The fact that you can choose $\|\mathbf{r}_i\|$, $\|\mathbf{s}_i\|$ such that the dot product is equal to 1 does not require that θ be zero. In fact, it usually will not be zero. It will be zero, however, if \mathbf{A} is symmetric. In that case, \mathbf{R} and \mathbf{S} become orthogonal matrices.

Step 3. Final justification that $\mathbf{A} = \mathbf{S} \mathbf{A} \mathbf{R}^T$. Finally, now that we have shown that $\mathbf{R}^T \mathbf{S} = \mathbf{I}_M$, we go back to Equations (5.12) and (5.47):

$$\mathbf{A} \mathbf{S} = \mathbf{S} \mathbf{A} \quad (5.12)$$

and

$$\mathbf{A}^T \mathbf{R} = \mathbf{R} \mathbf{A} \quad (5.47)$$

If we postmultiply Equation (5.12) by \mathbf{R}^T , we have

$$\mathbf{A} \mathbf{S} \mathbf{R}^T = \mathbf{S} \mathbf{A} \mathbf{R}^T \quad (5.70)$$

But

$$\mathbf{S}\mathbf{R}^T = \mathbf{I}_M \quad (5.71)$$

because, if we start with

$$\mathbf{R}^T\mathbf{S} = \mathbf{I}_M \quad (5.63)$$

which we so laboriously proved in the last pages, and postmultiply by \mathbf{S}^{-1} , we obtain

$$\mathbf{R}^T\mathbf{S}\mathbf{S}^{-1} = \mathbf{S}^{-1} \quad (5.72a)$$

or

$$\mathbf{R}^T = \mathbf{S}^{-1} \quad (5.72b)$$

Premultiplying by \mathbf{S} gives

$$\mathbf{S}\mathbf{R}^T = \mathbf{I}_M \quad (5.73)$$

as required. Therefore, at long last, we have

$$\boxed{\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{R}^T} \quad (5.74)$$

Equation (5.74) shows that an arbitrary square $M \times M$ matrix \mathbf{A} can be decomposed into the product of three matrices

$$\mathbf{S} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_M \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (5.11)$$

where \mathbf{S} is an $M \times M$ matrix, each column of which is an eigenvector \mathbf{s}_i such that

$$\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \quad (5.9)$$

$$\mathbf{R} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_M \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (5.43)$$

where \mathbf{R} is an $M \times M$ matrix, each column of which is an eigenvector \mathbf{r}_i such that

$$\mathbf{A}^T\mathbf{r}_i = \lambda_i\mathbf{r}_i \quad (5.38)$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_M \end{bmatrix} \quad (5.10)$$

where Λ is a diagonal $M \times M$ matrix with the eigenvalues λ_i of

$$\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \quad (5.9)$$

along the diagonal and zeros everywhere else.

A couple of points are worth noting before going on to use this theorem to find the inverse of \mathbf{A} , if it exists. First, not all of the eigenvalues λ_i are necessarily real. Some may be zero. Some may even be repeated. Second, note also that taking the transpose of Equation (5.74) yields

$$\mathbf{A}^T = \mathbf{R}\Lambda\mathbf{S}^T \quad (5.75)$$

5.4.4 Finding the Inverse \mathbf{A}^{-1} for the $M \times M$ Matrix \mathbf{A}

The goal in this section is to use Equation (5.74) to find \mathbf{A}^{-1} , the inverse to \mathbf{A} in the exact mathematical sense of

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_M \quad (5.76a)$$

and

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_M \quad (5.76b)$$

We start with Equation (5.9), the original statement of the eigenvalue problem for \mathbf{A}

$$\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \quad (5.9)$$

Premultiply by \mathbf{A}^{-1} (assuming, for the moment, that it exists)

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{A}\mathbf{s}_i &= \mathbf{A}^{-1}(\lambda_i\mathbf{s}_i) \\ &= \lambda_i\mathbf{A}^{-1}\mathbf{s}_i \end{aligned} \quad (5.77)$$

But, of course,

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_M$$

by Equation (5.76a). Therefore,

$$\mathbf{s}_i = \lambda_i\mathbf{A}^{-1}\mathbf{s}_i \quad (5.78)$$

or, rearranging terms

$$\mathbf{A}^{-1}\mathbf{s}_i = (1/\lambda_i)\mathbf{s}_i \quad \lambda_i \neq 0 \quad (5.79)$$

Equation (5.79) is of the form of an eigenvalue problem. In fact, it is a statement of the eigenvalue problem for \mathbf{A}^{-1} ! The eigenvalues for \mathbf{A}^{-1} are given by the reciprocal of the eigenvalues for \mathbf{A} . \mathbf{A} and \mathbf{A}^{-1} share the same eigenvectors s_i .

Since we know the eigenvalues and eigenvectors for \mathbf{A}^{-1} , we can use the information we have learned on how to decompose square matrices in Equation (5.74) to write \mathbf{A}^{-1} as

$$\boxed{\mathbf{A}^{-1} = \mathbf{S}\mathbf{\Lambda}^{-1}\mathbf{R}^T} \quad (5.80)$$

where

$$\mathbf{\Lambda}^{-1} = \begin{bmatrix} 1/\lambda_1 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_M \end{bmatrix} \quad (5.81)$$

Hence, if we can decompose \mathbf{A} as

$$\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{R}^T \quad (5.74)$$

one can find, if it exists, \mathbf{A}^{-1} as

$$\mathbf{A}^{-1} = \mathbf{S}\mathbf{\Lambda}^{-1}\mathbf{R}^T \quad (5.80)$$

Of course, if $\lambda_i = 0$ for any i , then $1/\lambda_i$ is undefined and hence \mathbf{A}^{-1} does not exist.

5.4.5 What Happens When There are Zero Eigenvalues?

Suppose that some of the eigenvalues λ_i of

$$\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i \quad (5.9)$$

are zero. What happens then? First, of course, \mathbf{A}^{-1} does not exist in the mathematical sense of Equations (5.76a–b). Given that, however, let us look in more detail.

First, suppose there are P nonzero λ_i and $(M - P)$ zero λ_i . We can order the λ_i such that

$$|\lambda_1| \geq |\lambda_2| \geq \cdots |\lambda_P| > 0 \quad (5.82)$$

and

$$\lambda_{P+1} = \lambda_{P+2} = \cdots = \lambda_M = 0 \quad (5.83)$$

Recall that one is always free to order the λ_i any way one chooses, as long as the \mathbf{s}_i and \mathbf{r}_i in Equation (5.74) are ordered the same way.

Then, we can rewrite $\mathbf{\Lambda}$ as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \\ & & \ddots & \\ \vdots & & & \lambda_P & \vdots \\ & & & & 0 & \ddots \\ 0 & \cdots & & & & 0 \end{bmatrix} \quad (5.84)$$

Consider Equation (5.74) again. We have that

$$\mathbf{A} = \mathbf{S}\Lambda\mathbf{R}^T \quad (5.74)$$

We can write out the right-hand side as

$$\begin{bmatrix} \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_P & \mathbf{s}_{P+1} & \cdots & \mathbf{s}_M \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \\ & & \ddots & \\ \vdots & & & \lambda_P \\ & & & & 0 & \ddots \\ 0 & \cdots & & & & 0 \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{r}_1 & \cdots \\ \cdots & \mathbf{r}_2 & \cdots \\ \cdots & \mathbf{r}_P & \cdots \\ \cdots & \mathbf{r}_{P+1} & \cdots \\ \vdots & & \vdots \\ \cdots & \mathbf{r}_M & \cdots \end{bmatrix} \quad (5.85)$$

where \mathbf{S} , Λ , and \mathbf{R} are all $M \times M$ matrices. Multiplying out $\mathbf{S}\Lambda$ explicitly yields

$$\begin{bmatrix} \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \lambda_1 \mathbf{s}_1 & \lambda_2 \mathbf{s}_2 & \cdots & \lambda_P \mathbf{s}_P & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{r}_1 & \cdots \\ \cdots & \mathbf{r}_2 & \cdots \\ \vdots & & \vdots \\ \cdots & \mathbf{r}_P & \cdots \\ \cdots & \mathbf{r}_{P+1} & \cdots \\ \vdots & & \vdots \\ \cdots & \mathbf{r}_M & \cdots \end{bmatrix} \begin{matrix} \overline{\uparrow} \\ P \\ \downarrow \\ \overline{\uparrow} \\ M-P \\ \downarrow \end{matrix} \quad (5.86)$$

$$|\leftarrow P \Rightarrow| \quad |\leftarrow M-P \Rightarrow| \quad |\leftarrow M \Rightarrow|$$

where we see that the last $(M - P)$ columns of the product $\mathbf{S}\Lambda$ are all zero. Note also that the last $(M - P)$ rows of \mathbf{R}^T (or, equivalently, the last $(M - P)$ columns of \mathbf{R}) will all be multiplied by zeros.

This means that $\mathbf{s}_{P+1}, \mathbf{s}_{P+2}, \dots, \mathbf{s}_M$ and $\mathbf{r}_P, \mathbf{r}_{P+1}, \dots, \mathbf{r}_M$ are not needed to form \mathbf{A} ! We say that these eigenvectors are obliterated in (or by) \mathbf{A} , or that \mathbf{A} is blind to them.

In order not to have to write out this partitioning in long hand each time, let us make the following definitions:

1. Let $\mathbf{S} = [\mathbf{S}_P | \mathbf{S}_0]$, where

$$\mathbf{S}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_P \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.87)$$

is an $M \times P$ matrix with the P eigenvectors of $\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i$ associated with the P nonzero eigenvalues λ_i as columns, and where

$$\mathbf{S}_0 = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{s}_{P+1} & \mathbf{s}_{P+2} & \cdots & \mathbf{s}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.88)$$

is an $M \times (M - P)$ matrix with the $(M - P)$ eigenvectors associated with the zero eigenvalues of $\mathbf{A}\mathbf{s}_i = \lambda_i\mathbf{s}_i$ as columns.

2. Let $\mathbf{R} = [\mathbf{R}_P | \mathbf{R}_0]$, where

$$\mathbf{R}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_P \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.89)$$

is an $M \times P$ matrix with the P eigenvectors of $\mathbf{A}^T\mathbf{r}_i = \lambda_i\mathbf{r}_i$ associated with the P nonzero eigenvalues λ_i as columns, and where

$$\mathbf{R}_0 = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{r}_{P+1} & \mathbf{r}_{P+2} & \cdots & \mathbf{r}_M \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (5.90)$$

is an $M \times (M - P)$ matrix with the $(M - P)$ eigenvectors associated with the zero eigenvalues of $\mathbf{A}^T\mathbf{r}_i = \lambda_i\mathbf{r}_i$ as columns.

3. Let

$$\Lambda_P = \begin{bmatrix} \Lambda_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where Λ_P is the $P \times P$ subset of Λ with the P nonzero eigenvalues λ_i along the diagonal and zeros elsewhere, as shown below

$$\Lambda_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_P \end{bmatrix} \quad (5.91)$$

and where the rest of Λ consists entirely of zeros.

We see, then, that Equation (5.86) implies that \mathbf{A} can be reconstructed with just \mathbf{S}_P , Λ_P , and \mathbf{R}_P as

$$\boxed{\begin{array}{rcccl} \mathbf{A} & = & \mathbf{S}_P & \Lambda_P & \mathbf{R}_P^T \\ M \times M & & M \times P & P \times P & P \times M \end{array}} \quad (5.92)$$

It is important to note that \mathbf{A} can be reconstructed using either Equation (5.74) or Equation (5.92). The benefit of using Equation (5.92) is that the matrices are smaller, and it could save you effort not to have to calculate the \mathbf{r}_i , \mathbf{s}_i associated with the zero eigenvalues. An important insight into the problem, however, is that although \mathbf{A} can be reconstructed without any information about directions associated with eigenvectors having zero eigenvalues, no information can be retrieved, or gained, about these directions in an inversion.

5.4.6 Some Notes on the Properties of \mathbf{S}_P and \mathbf{R}_P

1. At best, \mathbf{S}_P is semiorthogonal. It is possible that

$$\mathbf{S}_P^T \mathbf{S}_P = \mathbf{I}_P \quad (5.93)$$

depending on the form (or information) of \mathbf{A} . Note that the product $\mathbf{S}_P^T \mathbf{S}_P$ has dimension $(P \times M)(M \times P) = (P \times P)$, independent of M . The product will equal \mathbf{I}_P if and only if the P columns of \mathbf{S}_P are all orthogonal to one another.

It is never possible that

$$\mathbf{S}_P \mathbf{S}_P^T = \mathbf{I}_M \quad (5.94)$$

This product has dimension $(M \times P)(P \times M) = (M \times M)$. \mathbf{S}_P has M rows, but only P of them can be independent. Each row of \mathbf{S}_P can be thought of as a vector in P -space (since there are P columns). Only P vectors in P -space can be linearly independent, and $M > P$.

2. Similar arguments can be made about \mathbf{R}_P . That is, at best, \mathbf{R}_P is semiorthogonal. Thus

$$\mathbf{R}_P^T \mathbf{R}_P = \mathbf{I}_P \quad (5.95)$$

is possible, depending on the structure of \mathbf{A} . It is never possible that

$$\mathbf{R}_P \mathbf{R}_P^T = \mathbf{I}_M \quad (5.96)$$

3. \mathbf{A}^{-1} does not exist (since there are zero eigenvalues).
4. If there is some solution \mathbf{x} to $\mathbf{A}\mathbf{x} = \mathbf{b}$, (which is possible, even if \mathbf{A}^{-1} does not exist), then there are an infinite number of solutions. To see this, we note that

$$\mathbf{A}\mathbf{s}_i = \lambda_i \mathbf{s}_i = 0 \quad i = P + 1, P + 2, \dots, M \quad (5.97)$$

This means that if

$$\mathbf{Ax} = \mathbf{b} \quad (5.1)$$

for some \mathbf{x} , then if we add \mathbf{s}_i , $(P + 1) \leq i \leq M$, to \mathbf{x} , we have

$$\mathbf{A}[\mathbf{x} + \mathbf{s}_i] = \mathbf{b} + \lambda_i \mathbf{s}_i \quad (5.98a)$$

$$= \mathbf{b} \quad (5.98b)$$

since \mathbf{A} is a linear operator. This means that one could write a general solution as

$$\hat{\mathbf{x}} = \mathbf{x} + \sum_{i=P+1}^M \alpha_i \mathbf{s}_i \quad (5.99)$$

where α_i is an arbitrary weighting factor.

5.5 Eigenvector Structure of \mathbf{m}_{LS}

5.5.1 Square Symmetric \mathbf{A} Matrix With Nonzero Eigenvalues

Recall that the least squares problem can always be transformed into the normal equations form that involves square, symmetric matrices. If we start with

$$\mathbf{Gm} = \mathbf{d} \quad (5.100)$$

$$\mathbf{G}^T \mathbf{Gm} = \mathbf{G}^T \mathbf{d} \quad (5.101)$$

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (5.102)$$

Let $\mathbf{A} = \mathbf{G}^T \mathbf{G}$ and $\mathbf{b} = \mathbf{G}^T \mathbf{d}$. Then we have

$$\mathbf{Am} = \mathbf{b} \quad (5.103)$$

$$\mathbf{m}_{LS} = \mathbf{A}^{-1} \mathbf{b} \quad (5.104)$$

where \mathbf{A} is a square, symmetric matrix. Now, recall the decomposition theorem for square, symmetric matrices:

$$\mathbf{A} = \mathbf{S} \mathbf{\Lambda} \mathbf{S}^T \quad (5.105)$$

where \mathbf{S} is the $M \times M$ matrix with columns of eigenvectors of \mathbf{A} , and $\mathbf{\Lambda}$ is the $M \times M$ diagonal matrix with elements of eigenvalues of \mathbf{A} . Then, we have shown

$$\mathbf{A}^{-1} = \mathbf{S} \mathbf{\Lambda}^{-1} \mathbf{S}^T \quad (5.106)$$

and

$$\mathbf{m}_{LS} = \mathbf{S} \mathbf{\Lambda}^{-1} \mathbf{S}^T \mathbf{b} \quad (5.107)$$

Let's take a closer look at the "structure" of \mathbf{m}_{LS} . The easiest way to do this is to use a simple example with $M = 2$. Then

$$\mathbf{A}^{-1} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} 1/\lambda_1 & 0 \\ 0 & 1/\lambda_2 \end{bmatrix} \begin{bmatrix} s_{11} & s_{21} \\ s_{12} & s_{22} \end{bmatrix} \quad (5.108)$$

$$\mathbf{A}^{-1} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} \frac{s_{11}}{\lambda_1} & \frac{s_{12}}{\lambda_1} \\ \frac{s_{21}}{\lambda_1} & \frac{s_{22}}{\lambda_1} \end{bmatrix} \quad (5.109)$$

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{s_{11}^2}{\lambda_1} + \frac{s_{12}^2}{\lambda_2} & \frac{s_{11}s_{21}}{\lambda_1} + \frac{s_{12}s_{22}}{\lambda_2} \\ \frac{s_{21}s_{11}}{\lambda_1} + \frac{s_{22}s_{12}}{\lambda_2} & \frac{s_{21}^2}{\lambda_1} + \frac{s_{22}^2}{\lambda_2} \end{bmatrix} \quad (5.110)$$

This has the form of the sum of two matrices, one with $1/\lambda_1$ coefficient and the other with $1/\lambda_2$ coefficient.

$$\begin{aligned} \mathbf{A}^{-1} &= \frac{1}{\lambda_1} \begin{bmatrix} s_{11}^2 & s_{11}s_{21} \\ s_{21}s_{11} & s_{21}^2 \end{bmatrix} + \frac{1}{\lambda_2} \begin{bmatrix} s_{12}^2 & s_{12}s_{22} \\ s_{22}s_{12} & s_{22}^2 \end{bmatrix} \\ &= \frac{1}{\lambda_1} \begin{bmatrix} s_{11} \\ s_{21} \end{bmatrix} \begin{bmatrix} s_{11} & s_{21} \end{bmatrix} + \frac{1}{\lambda_2} \begin{bmatrix} s_{12} \\ s_{22} \end{bmatrix} \begin{bmatrix} s_{12} & s_{22} \end{bmatrix} \\ &= \frac{1}{\lambda_1} \mathbf{s}_1 \mathbf{s}_1^T + \frac{1}{\lambda_2} \mathbf{s}_2 \mathbf{s}_2^T \end{aligned} \quad (5.111)$$

In general, for a square symmetric matrix

$$\mathbf{A}^{-1} = \frac{1}{\lambda_1} \mathbf{s}_1 \mathbf{s}_1^T + \frac{1}{\lambda_2} \mathbf{s}_2 \mathbf{s}_2^T + \dots + \frac{1}{\lambda_M} \mathbf{s}_M \mathbf{s}_M^T \quad (5.112)$$

and

$$\mathbf{A} = \lambda_1 \mathbf{s}_1 \mathbf{s}_1^T + \lambda_2 \mathbf{s}_2 \mathbf{s}_2^T + \dots + \lambda_M \mathbf{s}_M \mathbf{s}_M^T \quad (5.113)$$

where \mathbf{s}_i is the i th eigenvector of \mathbf{A} and \mathbf{A}^{-1} .

Now, let's finish forming \mathbf{m}_{LS} for the simple 2×2 case.

$$\begin{aligned} \mathbf{m}_{LS} &= \mathbf{A}^{-1} \mathbf{b} \\ &= \frac{1}{\lambda_1} \mathbf{s}_1 \mathbf{s}_1^T \mathbf{b} + \frac{1}{\lambda_2} \mathbf{s}_2 \mathbf{s}_2^T \mathbf{b} \\ &= \frac{1}{\lambda_1} (s_{11}b_1 + s_{21}b_2) \mathbf{s}_1 + \frac{1}{\lambda_2} (s_{12}b_1 + s_{22}b_2) \mathbf{s}_2 \end{aligned} \quad (5.114)$$

Or, in general,

$$\begin{aligned}
 \mathbf{m}_{LS} &= \sum_i^M \frac{1}{\lambda_i} \left(\sum_j^M s_{ji} b_j \right) \mathbf{s}_i \\
 &= \sum_i \frac{1}{\lambda_i} (\mathbf{s}_i^T \mathbf{b}) \mathbf{s}_i \\
 &= \sum_i \frac{c_i}{\lambda_i} \mathbf{s}_i
 \end{aligned} \tag{5.115}$$

where $\mathbf{s}_i^T \mathbf{b} \equiv \mathbf{s}_i \cdot \mathbf{b}$ = the projection of data in the \mathbf{s}_i direction, and c_i are the constants.

In this form, we can see that the least squares solution of $\mathbf{A}\mathbf{m} = \mathbf{b}$ is composed of a weighted sum of the eigenvectors of \mathbf{A} . The coefficients of the eigenvectors are constants composed of two parts: one is the inverse of the corresponding eigenvalue, and the second is the projection of the data in the direction of the corresponding eigenvector. This form also clearly shows why there is no inverse if one or more of the eigenvalues of \mathbf{A} is zero and why very small eigenvalues can make \mathbf{m}_{LS} unstable. It also suggests how we might handle the case where \mathbf{A} has one or more zero eigenvalues.

5.5.2 The Case of Zero Eigenvalues

As we saw in section 5.4.5, we can order the eigenvalues from largest to smallest in absolute value, as long as the associated eigenvectors are ordered in the same way. Then we saw the remarkable result in Equation (5.92), that the matrix \mathbf{A} can be completely reconstructed using just the nonzero eigenvalues and eigenvectors.

$$|\lambda_1| > |\lambda_2| > |\lambda_i| \dots > |\lambda_p| > 0 \text{ and } \lambda_{(p+1)} = \dots = \lambda_M = 0 \tag{5.116}$$

That is,

$$\begin{aligned}
 \mathbf{A} &= \mathbf{S}_p \mathbf{\Lambda}_p \mathbf{S}_p^T \\
 &= \lambda_1 \mathbf{s}_1 \mathbf{s}_1^T + \dots + \lambda_p \mathbf{s}_p \mathbf{s}_p^T
 \end{aligned} \tag{5.117}$$

This suggests that perhaps we can construct \mathbf{A}^{-1} similarly using just the nonzero eigenvalues and their corresponding eigenvectors. Let's try,

$$\tilde{\mathbf{A}}^{-1} = \mathbf{S}_p \mathbf{\Lambda}_p^{-1} \mathbf{S}_p^T \tag{5.118}$$

which must, at least, exist. But, note that

$$\mathbf{A} \tilde{\mathbf{A}}^{-1} = \left[\mathbf{S}_p \mathbf{\Lambda}_p \mathbf{S}_p^T \right] \left[\mathbf{S}_p \mathbf{\Lambda}_p^{-1} \mathbf{S}_p^T \right] \tag{5.119}$$

Note that $\mathbf{S}_p^T \mathbf{S}_p = \mathbf{I}_p$ because \mathbf{A} is symmetric, and hence the eigenvectors are orthogonal, and $\mathbf{\Lambda}_p \mathbf{\Lambda}_p^{-1} = \mathbf{I}_p$, so

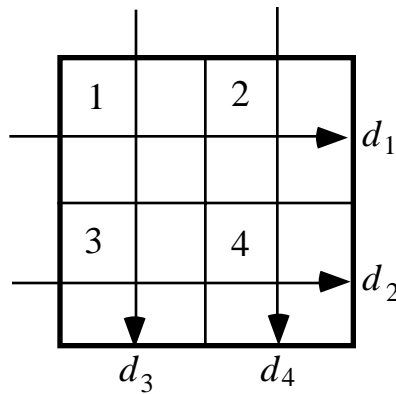
$$\mathbf{A} \tilde{\mathbf{A}}^{-1} = \mathbf{S}_p \mathbf{S}_p^T \neq \mathbf{I}_M \text{ (see 5.94)} \tag{5.120}$$

Like the case for \mathbf{A} , we cannot write a true inverse for \mathbf{A} using just the P nonzero eigenvectors—the true inverse does not exist. But, we can use $\tilde{\mathbf{A}}^{-1}$ as an "approximate" inverse for \mathbf{A} . Thus, in the case when \mathbf{A}^{-1} does not exist, we can use

$$\begin{aligned} \tilde{\mathbf{m}}_{LS} &= \tilde{\mathbf{A}}^{-1} \mathbf{b} \\ &= \mathbf{S}_P \mathbf{\Lambda}_P^{-1} \mathbf{S}_P^T \mathbf{b} \\ &= \sum_{i=1}^P \left(\frac{\mathbf{s}_i \cdot \mathbf{b}}{\lambda_i} \right) \mathbf{s}_i \end{aligned} \tag{5.121}$$

5.5.3 Simple Tomography Problem Revisited

The concepts developed in the previous section are best understood in the context of an example problem. Recall the simple tomography problem:



$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{5.122}$$

$$\mathbf{A} = \mathbf{G}^T \mathbf{G} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \tag{5.123}$$

$$\boldsymbol{\lambda} = \begin{bmatrix} 4 \\ 2 \\ 2 \\ 0 \end{bmatrix} \text{ (the eigenvalues of } \mathbf{A} \text{)} \tag{5.124}$$

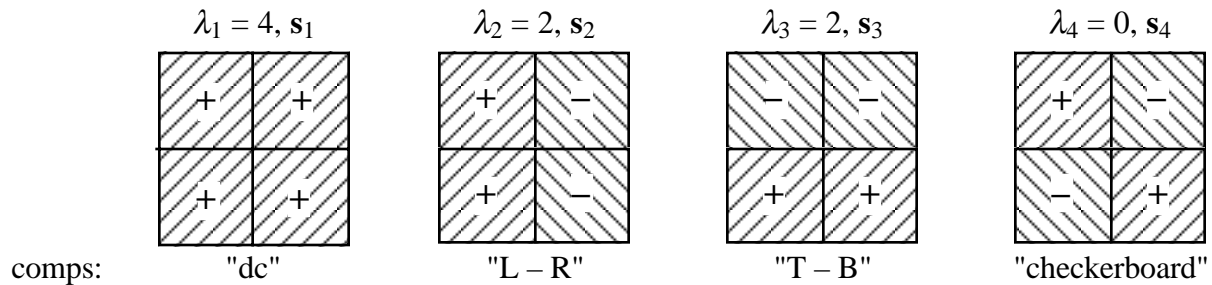
$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 \\ 0.5 & 0.5 & -0.5 & 0.5 \\ 0.5 & -0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & 0.5 \end{bmatrix} \quad (\text{the eigenvectors of } \mathbf{A}) \quad (5.125)$$

$$\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^T \quad (5.126)$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.127)$$

Let's look at the "patterns" in the eigenvectors, \mathbf{s}_i .



These "patterns" are the fundamental building blocks (basis functions) of all solutions in this four-space. Do you see why the eigenvalues correspond to their associated eigenvector patterns? Explain this in terms of the sampling of the four paths shooting through the medium. What other path(s) must we sample in order to make the zero eigenvalue nonzero?

Now, let's consider an actual model and the corresponding noise-free data. The model is

$$\mathbf{m} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{or graphically} \quad \begin{array}{|c|c|} \hline \text{diagonal lines} & 0 \\ \hline 0 & 0 \\ \hline \end{array} \quad (5.128)$$

The data are

$$\mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \mathbf{G}^T \mathbf{d} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (5.129)$$

First, \mathbf{A}^{-1} does not exist. But $\tilde{\mathbf{A}}^{-1} = \mathbf{S}_P \mathbf{\Lambda}_P^{-1} \mathbf{S}_P^T$ does, where $\mathbf{S}_P = [\mathbf{s}_1 \ \mathbf{s}_2 \ \mathbf{s}_3]$ and

$$\mathbf{\Lambda}_P^{-1} = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \quad (5.130)$$

then

$$\tilde{\mathbf{m}}_{LS} = \tilde{\mathbf{A}}^{-1} \mathbf{G}^T \mathbf{d} \quad (5.131)$$

We now have all the parts necessary to construct this solution. Let's use the alternate form of the solution:

$$\tilde{\mathbf{m}}_{LS} = \frac{1}{\lambda_1} [\mathbf{s}_1 \cdot \mathbf{b}] \mathbf{s}_1 + \frac{1}{\lambda_2} [\mathbf{s}_2 \cdot \mathbf{b}] \mathbf{s}_2 + \frac{1}{\lambda_3} [\mathbf{s}_3 \cdot \mathbf{b}] \mathbf{s}_3 \quad (5.132)$$

$$\mathbf{s}_1 \cdot \mathbf{b} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} = 1.0 + 0.5 + 0.5 = 2 \quad (5.133)$$

$$\mathbf{s}_2 \cdot \mathbf{b} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \\ -0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} = 1.0 - 0.5 + 0.5 = 1 \quad (5.134)$$

$$\mathbf{s}_3 \cdot \mathbf{b} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} = -1.0 - 0.5 + 0.5 = -1 \quad (5.135)$$

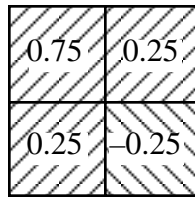
and note,

$$\mathbf{s}_4 \cdot \mathbf{b} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} = 1.0 - 0.5 - 0.5 = 0 \quad (5.136)$$

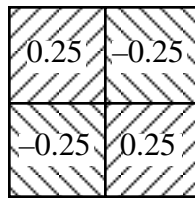
The data have no projection in the "null-space" \mathbf{s}_4 . The geometry of the experiment provides no constraints on the "checkerboard" pattern. How would we change the experiment to remedy this problem? Continuing the construction of the solution,

$$\begin{aligned} \tilde{\mathbf{m}}_{\text{LS}} &= \frac{2}{4} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \\ -0.5 \end{bmatrix} + \frac{-1}{2} \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \\ &= \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} + \begin{bmatrix} 0.25 \\ -0.25 \\ 0.25 \\ -0.25 \end{bmatrix} + \begin{bmatrix} 0.25 \\ 0.25 \\ -0.25 \\ -0.25 \end{bmatrix} = \begin{bmatrix} 0.75 \\ 0.25 \\ 0.25 \\ -0.25 \end{bmatrix} \end{aligned} \quad (5.134)$$

Graphically, this is



Does this solution fit the data? What do we need to add to this solution to get the true solution?



This is the eigenvector associated with the zero eigenvalues and represents the "null" space. Any scaled value of this pattern can be added and the data will not change—the data are "blind" to this pattern.

$$\tilde{\mathbf{m}}_{\text{LS}} = \begin{bmatrix} 0.75 \\ 0.25 \\ 0.25 \\ -0.25 \end{bmatrix} + \text{const.} \begin{bmatrix} 0.25 \\ -0.25 \\ -0.25 \\ 0.25 \end{bmatrix} \quad (5.135)$$

Summary

We now have all the tools to solve inverse problems, even those with zero eigenvalues! There are many real inverse problems that have been tackled using just what we have learned so far. In addition to truncating zero eigenvalues, the truncation method can be used to remove the effects of small eigenvalues. As you might expect, there is a "cost" or trade-off in truncating nonzero eigenvalues. Before we get into techniques to evaluate such trade-offs, we first turn to the generalization of the eigenvalue–eigenvector concepts to nonsquare matrices.

CHAPTER 6: SINGULAR-VALUE DECOMPOSITION (SVD)

6.1 Introduction

Having finished with the eigenvalue problem for $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} is square, we now turn our attention to the general $N \times M$ case $\mathbf{Gm} = \mathbf{d}$. First, the eigenvalue problem, per se, does not exist for $\mathbf{Gm} = \mathbf{d}$ unless $N = M$. This is because \mathbf{G} maps (transforms) a vector \mathbf{m} from M -space into a vector \mathbf{d} in N -space. The concept of “parallel” breaks down when the vectors lie in different dimensional spaces.

Since the eigenvalue problem is not defined for \mathbf{G} , we will try to construct a square matrix that includes \mathbf{G} (and, as it will turn out, \mathbf{G}^T) for which the eigenvalue problem is defined. This eigenvalue problem will lead us to *singular-value decomposition (SVD)*, a way to decompose \mathbf{G} into the product of three matrices (two eigenvector matrices \mathbf{V} and \mathbf{U} , associated with model and data spaces, respectively, and a singular-value matrix very similar to Λ from the eigenvalue problem for \mathbf{A}). Finally, it will lead us to the generalized inverse operator, defined in a way that is analogous to the inverse matrix to \mathbf{A} found using eigenvalue/eigenvector analysis.

The end result of SVD is

$$\boxed{\begin{matrix} \mathbf{G} & = & \mathbf{U}_P & \Lambda_P & \mathbf{V}_P^T \\ N \times M & & N \times P & P \times P & P \times M \end{matrix}} \quad (6.1)$$

where \mathbf{U}_P are the P N -dimensional eigenvectors of \mathbf{GG}^T , \mathbf{V}_P are the P M -dimensional eigenvectors of $\mathbf{G}^T\mathbf{G}$, and Λ_P is the $P \times P$ diagonal matrix with P singular values (positive square roots of the nonzero eigenvalues shared by \mathbf{GG}^T and $\mathbf{G}^T\mathbf{G}$) on the diagonal.

6.2 Formation of a New Matrix B

6.2.1 Formulating the Eigenvalue Problem With G

The way to construct an eigenvalue problem that includes \mathbf{G} is to form a square $(N + M) \times (N + M)$ matrix \mathbf{B} partitioned as follows:

$$\mathbf{B} = \left[\begin{array}{cc|c} \mathbf{0} & \mathbf{G} & \updownarrow \\ N \times N & N \times M & N \\ \hline \mathbf{G}^T & \mathbf{0} & \updownarrow \\ M \times N & M \times M & M \\ \hline \leftarrow N \Rightarrow & \leftarrow M \Rightarrow & \end{array} \right] \quad (6.2)$$

\mathbf{B} is *Hermitian* because

$$\mathbf{B}^T = \mathbf{B} \quad (6.3)$$

Note, for example,

$$B_{1,N+3} = G_{13} \quad (6.4a)$$

and

$$B_{N+3,1} = (\mathbf{G}^T)_{31} = G_{13}, \text{ etc.} \quad (6.4b)$$

6.2.2 The Role of \mathbf{G}^T as an Operator

Analogous to Equation (1.13), we can define an equation for \mathbf{G}^T as follows:

$$\begin{array}{ccc} \mathbf{G}^T & \mathbf{y} & = \mathbf{c} \\ M \times N & N \times 1 & M \times 1 \end{array} \quad (6.5)$$

We do not have to have a particular \mathbf{y} and \mathbf{c} in mind when we do this. We are simply interested in the mapping of an N -dimensional vector into an M -dimensional vector by \mathbf{G}^T .

We can combine $\mathbf{G}\mathbf{m} = \mathbf{d}$ and $\mathbf{G}^T\mathbf{y} = \mathbf{c}$, using \mathbf{B} , as

$$\left[\begin{array}{c|c} \mathbf{0} & \mathbf{G} \\ \hline \mathbf{G}^T & \mathbf{0} \end{array} \right] \left[\begin{array}{c} \mathbf{y} \\ \mathbf{m} \end{array} \right] = \left[\begin{array}{c} \mathbf{d} \\ \mathbf{c} \end{array} \right] \quad (6.6)$$

or

$$\begin{array}{ccc} \mathbf{B} & \mathbf{z} & = \mathbf{b} \\ (N+M) \times (N+M) & (N+M) \times 1 & (N+M) \times 1 \end{array} \quad (6.7)$$

where we have

$$\mathbf{z} = \left[\begin{array}{c} \mathbf{y} \\ \mathbf{m} \end{array} \right] \quad (6.8)$$

and

$$\mathbf{b} = \left[\begin{array}{c} \mathbf{d} \\ \mathbf{c} \end{array} \right] \quad (6.9)$$

Note that \mathbf{z} and \mathbf{b} are both $(N+M) \times 1$ column vectors.

6.3 The Eigenvalue Problem for \mathbf{B}

The eigenvalue problem for the $(N + M) \times (N + M)$ matrix \mathbf{B} is given by

$$\mathbf{B}\mathbf{w}_i = \eta_i \mathbf{w}_i \quad i = 1, 2, \dots, N + M \quad (6.10)$$

6.3.1 Properties of \mathbf{B}

The matrix \mathbf{B} is Hermitian. Therefore, all $N + M$ eigenvalues η_i are real. In preparation for solving the eigenvalue problem, we define the eigenvector matrix \mathbf{W} for \mathbf{B} as follows:

$$\begin{array}{c} \mathbf{W} \\ (N + M) \times (N + M) \end{array} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_{N+M} \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (6.11)$$

We note that \mathbf{W} is an orthogonal matrix, and thus

$$\mathbf{W}^T \mathbf{W} = \mathbf{W} \mathbf{W}^T = \mathbf{I}_{N+M} \quad (6.12)$$

This is equivalent to

$$\mathbf{w}_i^T \mathbf{w}_j = \delta_{ij} \quad (6.13)$$

where \mathbf{w}_i is the i th eigenvector in \mathbf{W} .

6.3.2 Partitioning \mathbf{W}

Each eigenvector \mathbf{w}_i is $(N + M) \times 1$. Consider partitioning \mathbf{w}_i such that

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \begin{array}{c} \overline{\uparrow} \\ N \\ \overline{\downarrow} \\ \overline{\uparrow} \\ M \\ \overline{\downarrow} \end{array} \quad (6.14)$$

That is, we “stack” an N -dimensional vector \mathbf{u}_i and an M -dimensional vector \mathbf{v}_i into a single $(N + M)$ -dimensional vector.

Then the eigenvalue problem for \mathbf{B} from Equation (6.10) becomes

$$\left[\begin{array}{c|c} \mathbf{0} & \mathbf{G} \\ \hline \mathbf{G}^T & \mathbf{0} \end{array} \right] \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} = \eta_i \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \quad (6.15)$$

This can be written as

$$\boxed{\begin{array}{ccc} \mathbf{G} & \mathbf{v}_i = \eta_i \mathbf{u}_i & i = 1, 2, \dots, N + M \\ N \times M & M \times 1 & N \times 1 \end{array}} \quad (6.16)$$

and

$$\boxed{\begin{array}{ccc} \mathbf{G}^T & \mathbf{u}_i = \eta_i \mathbf{v}_i & i = 1, 2, \dots, N + M \\ M \times N & N \times 1 & M \times 1 \end{array}} \quad (6.17)$$

Equations (6.16) and (6.17) together are called the *shifted eigenvalue problem for \mathbf{G}* . It is not an eigenvalue problem for \mathbf{G} , since \mathbf{G} is not square and eigenvalue problems are only defined for square matrices. Still, it is analogous to an eigenvalue problem. Note that \mathbf{G} operates on an M -dimensional vector and returns an N -dimensional vector. \mathbf{G}^T operates on an N -dimensional vector and returns an M -dimensional vector. Furthermore, the vectors are shared by \mathbf{G} and \mathbf{G}^T .

6.4 Solving the Shifted Eigenvalue Problem

Equations (6.16) and (6.17) can be solved by combining them into two related eigenvalue problems involving $\mathbf{G}^T \mathbf{G}$ and $\mathbf{G} \mathbf{G}^T$, respectively.

6.4.1 The Eigenvalue Problem for $\mathbf{G}^T \mathbf{G}$

Eigenvalue problems are only defined for square matrices. Note, then, that $\mathbf{G}^T \mathbf{G}$ is $M \times M$, and hence has an eigenvalue problem. The procedure is as follows:

Starting with Equation (6.17)

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i \quad (6.17)$$

Multiply both sides by η_i

$$\eta_i \mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{v}_i \quad (6.18)$$

or

$$\mathbf{G}^T(\eta_i \mathbf{u}_i) = \eta_i^2 \mathbf{v}_i \quad (6.19)$$

But, by Equation (6.16), we have

$$\mathbf{G}\mathbf{v}_i = \eta_i \mathbf{u}_i \quad (6.16)$$

Thus

$$\boxed{\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad i = 1, 2, \dots, M} \quad (6.20)$$

This is just the eigenvalue problem for $\mathbf{G}^T \mathbf{G}$! We were able to manipulate the shifted eigenvalue problem into an eigenvalue problem that, presumably, we can solve.

We make the following notes:

1. $\mathbf{G}^T \mathbf{G}$ is Hermitian.

$$(\mathbf{G}^T \mathbf{G})_{ij} = \sum_{k=1}^N (\mathbf{G}^T)_{ik} G_{kj} = \sum_{k=1}^N g_{ki} g_{kj} \quad (6.21)$$

$$(\mathbf{G}^T \mathbf{G})_{ji} = \sum_{k=1}^N (\mathbf{G}^T)_{jk} G_{ki} = \sum_{k=1}^N g_{kj} g_{ki} = (\mathbf{G}^T \mathbf{G})_{ij} \quad (6.22)$$

2. Therefore, all $M \eta_i^2$ are real. Because the diagonal entries of $\mathbf{G}^T \mathbf{G}$ are all ≥ 0 , then all η_i^2 are also ≥ 0 . This means that $\mathbf{G}^T \mathbf{G}$ is positive semidefinite (one definition of which is, simply, that all the eigenvalues are real and ≥ 0).

We can combine the M equations implied by Equation (6.20) into matrix notation as

$$\begin{matrix} \mathbf{G}^T \mathbf{G} & \mathbf{V} & = & \mathbf{V} & \mathbf{M} \\ M \times M & M \times M & & M \times M & M \times M \end{matrix} \quad (6.23)$$

where \mathbf{V} is defined as follows:

$$\mathbf{V} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_M \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (6.24)$$

$M \times M$

and

$$\mathbf{M} = \begin{bmatrix} \eta_1^2 & 0 & \cdots & 0 \\ 0 & \eta_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \eta_M^2 \end{bmatrix} \quad (6.25)$$

$M \times M$

3. Because $\mathbf{G}^T\mathbf{G}$ is a Hermitian matrix, \mathbf{V} is itself an orthogonal matrix:

$$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}_M \quad (6.26)$$

6.4.2 The Eigenvalue Problem for $\mathbf{G}\mathbf{G}^T$

The procedure for forming the eigenvalue problem for $\mathbf{G}\mathbf{G}^T$ is very analogous to that of $\mathbf{G}^T\mathbf{G}$. We note that $\mathbf{G}\mathbf{G}^T$ is $N \times N$. Starting with Equation (6.16),

$$\mathbf{G}\mathbf{v}_i = \eta_i \mathbf{u}_i \quad (6.16)$$

Again, multiply by η_i

$$\mathbf{G}(\eta_i \mathbf{v}_i) = \eta_i^2 \mathbf{u}_i \quad (6.27)$$

But by Equation (6.17), we have

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i \quad (6.17)$$

Thus

$$\boxed{\mathbf{G}\mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad i = 1, 2, \dots, N} \quad (6.28)$$

We make the following notes for this eigenvalue problem:

1. $\mathbf{G}\mathbf{G}^T$ is Hermitian.
2. $\mathbf{G}\mathbf{G}^T$ is positive semidefinite.
3. Combining the N equations in Equation (6.28), we have

$$\mathbf{G}\mathbf{G}^T \mathbf{U} = \mathbf{U}\mathbf{N} \quad (6.29)$$

where

$$\mathbf{U} = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (6.30)$$

$N \times N$

and

$$\mathbf{N} = \begin{bmatrix} \eta_1^2 & 0 & \cdots & 0 \\ 0 & \eta_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \eta_N^2 \end{bmatrix} \quad (6.31)$$

$N \times N$

4. \mathbf{U} is an orthogonal matrix

$$\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}_N \quad (6.32)$$

6.5 How Many η_i Are There, Anyway??

A careful look at Equations (6.10), (6.20), and (6.28) shows that the eigenvalue problems for \mathbf{B} , $\mathbf{G}^T \mathbf{G}$, and $\mathbf{G} \mathbf{G}^T$ are defined for $(N + M)$, M , and N values of i , respectively. Just how many η_i are there?

6.5.1 Introducing P , the Number of Nonzero Pairs $(+\eta_i, -\eta_i)$

Equation (6.10)

$$\mathbf{B} \mathbf{w}_i = \eta_i \mathbf{w}_i \quad (6.10)$$

can be used to determine $(N + M)$ real η_i . Equation (6.20),

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad (6.20)$$

can be used to determine M real η_i^2 since $\mathbf{G}^T \mathbf{G}$ is $M \times M$. Equation (6.28)

$$\mathbf{G} \mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad (6.28)$$

can be used to determine N real η_i^2 since $\mathbf{G} \mathbf{G}^T$ is $N \times N$.

This section will convince you, I hope, that the following are true:

1. There are P pairs of nonzero η_i , where each pair consists of $(+\eta_i, -\eta_i)$.
2. If $+\eta_i$ is an eigenvalue of

$$\mathbf{B} \mathbf{w}_i = \eta_i \mathbf{w}_i \quad (6.10)$$

and the associated eigenvector \mathbf{w}_i is given by

$$\mathbf{w}_i = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \quad (6.33)$$

then the eigenvector associated with $-\eta_i$ is given by

$$\mathbf{w}'_i = \begin{bmatrix} -\mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \quad (6.34)$$

3. There are $(N + M) - 2P$ zero η_i .
4. You can know everything you need to know about the shifted eigenvalue problem by retaining *only* the information associated with the positive η_i .
5. P is less than or equal to the minimum of N and M .

$$P \leq \min(N, M) \quad (6.35)$$

6.5.2 Finding the Eigenvector Associated With $-\eta_i$

Suppose that you have found \mathbf{w}_i , a solution to Equation (6.10) associated with η_i . It also satisfies the shifted eigenvalue problem

$$\mathbf{G}\mathbf{v}_i = \eta_i\mathbf{u}_i \quad (6.16)$$

and

$$\mathbf{G}^T\mathbf{u}_i = \eta_i\mathbf{v}_i \quad (6.17)$$

Let us try $-\eta_i$ as an eigenvalue and \mathbf{w}'_i given by

$$\mathbf{w}'_i = \begin{bmatrix} -\mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \quad (6.34)$$

and see if it satisfies Equations (6.16) and (6.17)

$$\mathbf{G}\mathbf{v}_i = (-\eta_i)(-\mathbf{u}_i) = \eta_i\mathbf{u}_i \quad (6.36)$$

and

$$\mathbf{G}^T(-\mathbf{u}_i) = (-\eta_i)\mathbf{v}_i \quad (6.37a)$$

or

$$\mathbf{G}^T\mathbf{u}_i = \eta_i\mathbf{v}_i \quad (6.37b)$$

From this we conclude that the nonzero eigenvalues of \mathbf{B} come in pairs. The relationship between the solutions is given in Equations (6.33) and (6.34). *Note that this property of paired eigenvalues and eigenvectors is not the case for the general eigenvalue problem.* It results from the symmetry of the shifted eigenvalue problem.

6.5.3 No New Information From the $-\eta_i$ System

Let us form an ordered eigenvalue matrix \mathbf{D} for \mathbf{B} given by

$$\mathbf{D} = \begin{bmatrix} \eta_1 & 0 & & \cdots & & & & & & 0 \\ 0 & \eta_2 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & \eta_P & & & & & & \\ & & & & -\eta_1 & & & & & \\ \vdots & & & & & -\eta_2 & & & & \vdots \\ & & & & & & \ddots & & & \\ & & & & & & & -\eta_P & & \\ & & & & & & & & 0 & \\ & & & & & & & & & \ddots \\ 0 & & & & \cdots & & & & & 0 \end{bmatrix} \quad (6.38)$$

$(N + M) \times (N + M)$

where $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_P$. Note that the ordering of matrices in eigenvalue problems is arbitrary, but must be internally consistent. Then the eigenvalue problem for \mathbf{B} from Equation (6.10) becomes

$$\mathbf{B}\mathbf{W} = \mathbf{W}\mathbf{D} \quad (6.39)$$

where now the $(N + M) \times (N + M)$ dimensional matrix \mathbf{W} is given by

$$\mathbf{W} = \begin{bmatrix} \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_P & -\mathbf{u}_1 & -\mathbf{u}_2 & \cdots & -\mathbf{u}_P & \mathbf{u}_{2P+1} & \cdots & \mathbf{u}_{N+M} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \hline \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P & \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P & \mathbf{v}_{2P+1} & \cdots & \mathbf{v}_{N+M} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} \quad (6.40)$$

$|\leftarrow P \Rightarrow| |\leftarrow P \Rightarrow| |\leftarrow (N + M) - 2P \Rightarrow|$

The second P eigenvectors certainly contain independent information about the eigenvectors \mathbf{w}_i in $(N + M)$ -space. They contain *no new* information, however, about \mathbf{u}_i or \mathbf{v}_i , in N - and M -space, respectively, since $-\mathbf{u}_i$ contains no information not already contained in $+\mathbf{u}_i$.

6.5.4 What About the Zero Eigenvalues η_i 's, $i = (2P + 1), \dots, N + M$?

For the zero eigenvalues, the shifted eigenvalue problem becomes

$$\mathbf{G}\mathbf{v}_i = \eta_i \mathbf{u}_i = \mathbf{0}_{N \times 1} \quad i = (2P + 1), \dots, (N + M) \quad (6.41a)$$

and

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i = \mathbf{0}_{M \times 1} \quad i = (2P + 1), \dots, (N + M) \quad (6.41b)$$

where $\mathbf{0}$ is a vector of zeros of the appropriate dimension.

If you premultiply Equation (6.41a) by \mathbf{G}^T and Equation (6.41b) by \mathbf{G} , you obtain

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \mathbf{G}^T \mathbf{0} = \mathbf{0}_{(M \times 1)} \quad (6.42)$$

and

$$\mathbf{G} \mathbf{G}^T \mathbf{u}_i = \mathbf{G} \mathbf{0} = \mathbf{0}_{(N \times 1)} \quad (6.43)$$

Therefore, we conclude that the \mathbf{u}_i , \mathbf{v}_i associated with zero η_i for \mathbf{B} are simply the eigenvectors of $\mathbf{G} \mathbf{G}^T$ and $\mathbf{G}^T \mathbf{G}$ associated with zero eigenvalues for $\mathbf{G} \mathbf{G}^T$ and $\mathbf{G}^T \mathbf{G}$, respectively!

6.5.5 How Big is P?

Now that we have seen that the eigenvalues come in P pairs of nonzero values, how can we determine the size of P ? We will see that you can determine P from either $\mathbf{G}^T \mathbf{G}$ or $\mathbf{G} \mathbf{G}^T$, and that P is bounded by the smaller of N and M , the number of observations and model parameters, respectively. The steps are as follows.

Step 1. Let the number of nonzero eigenvalues η_i^2 of $\mathbf{G}^T \mathbf{G}$ be P . Since $\mathbf{G}^T \mathbf{G}$ is $M \times M$, there are only M η_i^2 all together. Thus, P is less than or equal to M .

Step 2. If $\eta_i^2 \neq 0$ is an eigenvalue of $\mathbf{G}^T \mathbf{G}$, then it is also an eigenvalue of $\mathbf{G} \mathbf{G}^T$ since

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad (6.20)$$

and

$$\mathbf{G} \mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad (6.28)$$

Thus the nonzero η_i 's are shared by $\mathbf{G}^T \mathbf{G}$ and $\mathbf{G} \mathbf{G}^T$.

Step 3. P is less than or equal to N since $\mathbf{G} \mathbf{G}^T$ is $N \times N$. Therefore, since $P \leq M$ and $P \leq N$,

$$P \leq \min(N, M)$$

Thus, to determine P , you can do the eigenvalue problem for either $\mathbf{G}^T\mathbf{G}$ ($M \times M$) or $\mathbf{G}\mathbf{G}^T$ ($N \times N$). It makes sense to choose the smaller of the two matrices. That is, one chooses $\mathbf{G}^T\mathbf{G}$ if $M < N$, or $\mathbf{G}\mathbf{G}^T$ if $N < M$.

6.6 Introducing Singular Values

6.6.1 Introduction

Recalling Equation (6.29) defining the eigenvalue problem for $\mathbf{G}\mathbf{G}^T$

$$\begin{matrix} \mathbf{G}\mathbf{G}^T & \mathbf{U} & = & \mathbf{U} & \mathbf{N} \\ N \times N & N \times N & & N \times N & N \times N \end{matrix} \quad (6.29)$$

The matrix \mathbf{U} contains the eigenvectors \mathbf{u}_i , and can be ordered as

$$\begin{matrix} \mathbf{U} \\ N \times N \end{matrix} = \begin{matrix} \left[\begin{array}{ccc|ccc} \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_P & \mathbf{u}_{P+1} & \cdots & \mathbf{u}_N \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \end{array} \right] \begin{array}{l} \overleftarrow{\uparrow} \\ N \\ \overrightarrow{\downarrow} \end{array} \\ \left| \leftarrow \quad P \quad \Rightarrow \right| \left| \leftarrow \quad N-P \quad \Rightarrow \right| \end{matrix} \quad (6.44)$$

or

$$\mathbf{U} = [\mathbf{U}_P \mid \mathbf{U}_0] \quad (6.45)$$

Recall Equation (6.23), which defined the eigenvalue problem for $\mathbf{G}^T\mathbf{G}$,

$$\begin{matrix} \mathbf{G}^T\mathbf{G} & \mathbf{V} & = & \mathbf{V} & \mathbf{M} \\ M \times M & M \times M & & M \times M & M \times M \end{matrix} \quad (6.23)$$

The matrix \mathbf{V} of eigenvectors is given by

$$\begin{matrix} \mathbf{V} \\ M \times M \end{matrix} = \begin{matrix} \left[\begin{array}{ccc|ccc} \vdots & \vdots & & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P & \mathbf{v}_{P+1} & \cdots & \mathbf{v}_M \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \end{array} \right] \begin{array}{l} \overleftarrow{\uparrow} \\ M \\ \overrightarrow{\downarrow} \end{array} \\ \left| \leftarrow \quad P \quad \Rightarrow \right| \left| \leftarrow \quad M-P \quad \Rightarrow \right| \end{matrix} \quad (6.46)$$

or

$$\mathbf{V} = [\mathbf{V}_P \mid \mathbf{V}_0] \quad (6.47)$$

where the $\mathbf{u}_i, \mathbf{v}_i$ satisfy

$$\mathbf{G}\mathbf{v}_i = \eta_i \mathbf{u}_i \quad (6.16)$$

and

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i \tag{6.17}$$

and where we have chosen the P positive η_i from

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \tag{6.20}$$

$$\mathbf{G} \mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \tag{6.28}$$

Note that it is customary to order the $\mathbf{u}_i, \mathbf{v}_i$ such that

$$\eta_1 \geq \eta_2 \geq \dots \geq \eta_P \tag{6.48}$$

6.6.2 Definition of the Singular Value

We define a singular value λ_i from Equation (6.20) or (6.28) as the positive square root of the eigenvalue η_i^2 for $\mathbf{G}^T \mathbf{G}$ or $\mathbf{G} \mathbf{G}^T$. That is,

$$\lambda_i = +\sqrt{\eta_i^2} \tag{6.49}$$

Singular values are not eigenvalues. λ_i is not an eigenvalue for \mathbf{G} or \mathbf{G}^T , since the eigenvalue problem is not defined for \mathbf{G} or \mathbf{G}^T , $N \neq M$. They are, of course, eigenvalues for \mathbf{B} in Equation (6.10), but we will never explicitly deal with \mathbf{B} . The matrix \mathbf{B} is a construct that allowed us to formulate the shifted eigenvalue problem, but in practice, it is never formed. Nevertheless, you will often read, or hear, λ_i referred to as an eigenvalue.

6.6.3 Definition of Λ , the Singular-Value Matrix

We can form an $N \times M$ matrix with the singular values on the diagonal. If $M > N$, it has the form

$$\Lambda_{N \times M} = \left[\begin{array}{cccc|ccc} \lambda_1 & 0 & \dots & 0 & & & \\ 0 & \lambda_2 & & & & & \\ & & \ddots & & & & \\ \vdots & & & \lambda_P & \vdots & & \mathbf{0} \\ & & & & 0 & \dots & \\ 0 & \dots & & & & \ddots & 0 \end{array} \right] \begin{array}{l} \uparrow \\ \\ \\ \downarrow \\ \\ \downarrow \end{array} \tag{6.50a}$$

$\leftarrow \quad N \quad \Rightarrow \leftarrow M - N \Rightarrow$

If $N > M$, it has the form (next page)

$$\Lambda = \begin{array}{c} N \times M \\ \left[\begin{array}{cccc} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & \ddots & \\ & & & \lambda_P \\ & & & & 0 \\ & & & & & \ddots \\ 0 & & \dots & & & 0 \end{array} \right] \begin{array}{l} \overline{\uparrow} \\ \\ \vdots \\ M \\ \\ \downarrow \\ \overline{\uparrow} \\ N - M \\ \downarrow \end{array} \end{array} \quad (6.50b)$$

$$\left| \leftarrow \quad \quad \quad \Rightarrow \right|$$

Then the shifted eigenvalue problem

$$\mathbf{G}\mathbf{v}_i = \eta_i \mathbf{u}_i \quad (6.16)$$

and

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i \quad (6.17)$$

can be written as

$$\mathbf{G}\mathbf{v}_i = \lambda_i \mathbf{u}_i \quad (6.51)$$

and

$$\mathbf{G}^T \mathbf{u}_i = \lambda_i \mathbf{v}_i \quad (6.52)$$

where η_i has been replaced by λ_i since all information about \mathbf{U} , \mathbf{V} can be obtained from the positive η_i .

Equations (6.51) and (6.52) can be written in matrix notation as

$$\boxed{
 \begin{array}{cccc}
 \mathbf{G} & \mathbf{V} & = & \mathbf{U} & \Lambda \\
 N \times M & M \times M & & N \times N & N \times M
 \end{array}
 } \quad (6.53)$$

and

$$\boxed{
 \begin{array}{cccc}
 \mathbf{G}^T & \mathbf{U} & = & \mathbf{V} & \Lambda^T \\
 M \times N & N \times N & & M \times M & M \times N
 \end{array}
 } \quad (6.54)$$

6.7 Derivation of the Fundamental Decomposition Theorem for General \mathbf{G} ($N \times M, N \neq M$)

Recall that we used the eigenvalue problem for square \mathbf{A} and \mathbf{A}^T to derive a decomposition theorem for square matrices:

$$\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{R}^T \tag{5.74}$$

where \mathbf{S} , \mathbf{R} , and $\mathbf{\Lambda}$ are eigenvector and eigenvalue matrices associated with \mathbf{A} and \mathbf{A}^T . We are now ready to derive an analogous decomposition theorem for the general $N \times M, N \neq M$ matrix \mathbf{G} .

We start with Equation (6.53)

$$\begin{matrix} \mathbf{G} & \mathbf{V} & = & \mathbf{U} & \mathbf{\Lambda} \\ N \times M & M \times M & & N \times N & N \times M \end{matrix} \tag{6.53}$$

postmultiply by \mathbf{V}^T

$$\mathbf{G}\mathbf{V}\mathbf{V}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \tag{6.55}$$

But \mathbf{V} is an orthogonal matrix. That is,

$$\mathbf{V}\mathbf{V}^T = \mathbf{I}_M \tag{6.26}$$

since $\mathbf{G}^T\mathbf{G}$ is Hermitian, and the eigenvector matrices of Hermitian matrices are orthogonal. Therefore, we have the fundamental decomposition theorem for a general matrix given by

$$\boxed{\begin{matrix} \mathbf{G} & = & \mathbf{U} & \mathbf{\Lambda} & \mathbf{V}^T \\ N \times M & & N \times N & N \times M & M \times M \end{matrix}} \tag{6.56}$$

By taking the transpose of Equation (6.56), we obtain also

$$\boxed{\begin{matrix} \mathbf{G}^T & = & \mathbf{V} & \mathbf{\Lambda}^T & \mathbf{U}^T \\ M \times N & & M \times M & M \times N & N \times N \end{matrix}} \tag{6.57}$$

where

$$\mathbf{U}_{N \times N} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_p & \mathbf{u}_{p+1} & \cdots & \mathbf{u}_N \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} = [\mathbf{U}_p \mid \mathbf{U}_0] \tag{6.58}$$

and

$$\mathbf{V}_{M \times M} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \cdots & \mathbf{v}_P & \mathbf{v}_{P+1} & \cdots & \mathbf{v}_M \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} = [\mathbf{V}_P \mid \mathbf{V}_0] \quad (6.59)$$

and

$$\mathbf{\Lambda}_{N \times M} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \\ & & \ddots & \\ \vdots & & & \lambda_P & \vdots \\ & & & & 0 & \\ & & & & & \ddots \\ 0 & \cdots & & & & 0 \end{bmatrix} \quad (6.60)$$

6.8 Singular-Value Decomposition (SVD)

6.8.1 Derivation of Singular-Value Decomposition

We will see below that \mathbf{G} can be decomposed without any knowledge of the parts of \mathbf{U} or \mathbf{V} associated with zero singular values $\lambda_i, i > P$. We start with the fundamental decomposition theorem

$$\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (6.56)$$

Let us introduce a $P \times P$ singular-value matrix $\mathbf{\Lambda}_P$ that is a subset of $\mathbf{\Lambda}$:

$$\mathbf{\Lambda}_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_P \end{bmatrix} \quad (6.61)$$

We now write out Equation (6.56) in terms of the partitioned matrices as

$$\mathbf{G} = N \begin{array}{c} \overline{\uparrow} \\ \downarrow \\ \left[\begin{array}{c|c} & \\ \hline & \end{array} \right] \\ \leftarrow P \Rightarrow \leftarrow N-P \Rightarrow \end{array} \begin{array}{c} \overline{\uparrow} \\ \downarrow \\ \overline{\uparrow} \\ \downarrow \\ \overline{\uparrow} \\ \downarrow \end{array} \begin{array}{c} P \\ N-P \\ M-P \end{array} \begin{array}{c} \left[\begin{array}{c|c} \mathbf{\Lambda}_P & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] \\ \leftarrow P \Rightarrow \leftarrow M-P \Rightarrow \end{array} \begin{array}{c} \left[\begin{array}{c} \mathbf{V}_P^T \\ \mathbf{V}_0^T \end{array} \right] \\ \leftarrow M \Rightarrow \end{array} \begin{array}{c} \overline{\uparrow} \\ \downarrow \\ \overline{\uparrow} \\ \downarrow \end{array} \begin{array}{c} P \\ M-P \end{array} \quad (6.62)$$

$$= N \left[\begin{array}{c|c} \uparrow & \\ \mathbf{U}_P \Lambda_P & \mathbf{0} \\ \downarrow & \\ \leftarrow P \Rightarrow & \leftarrow M-P \Rightarrow \end{array} \right] \begin{bmatrix} \mathbf{V}_P^T \\ \mathbf{V}_0^T \end{bmatrix} \quad (6.63)$$

$$= \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \quad (6.64)$$

That is, we can write \mathbf{G} as

$$\boxed{\begin{array}{ccccc} \mathbf{G} & = & \mathbf{U}_P & \Lambda_P & \mathbf{V}_P^T \\ N \times M & & N \times P & P \times P & P \times M \end{array}} \quad (6.65)$$

Equation (6.65) is known as the *Singular-Value Decomposition Theorem for \mathbf{G}* .

The matrices in Equation (6.65) are

1. \mathbf{G} = an arbitrary $N \times M$ matrix.
2. The eigenvector matrix \mathbf{U}

$$\mathbf{U}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_P \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (6.66)$$

where \mathbf{u}_i are the P N -dimensional eigenvectors of

$$\mathbf{G}\mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad (6.28)$$

associated with nonzero singular values λ_i .

3. The eigenvector matrix \mathbf{V}

$$\mathbf{V}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (6.67)$$

where \mathbf{v}_i are the P M -dimensional eigenvectors of

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad (6.20)$$

associated with nonzero singular values λ_i , and

4. The singular-value matrix Λ

$$\Lambda_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_P \end{bmatrix} \quad (6.61)$$

where λ_i is the nonzero singular value associated with \mathbf{u}_i and \mathbf{v}_i , $i = 1, \dots, P$.

6.8.2 Rewriting the Shifted Eigenvalue Problem

Now that we have seen that we can reconstruct \mathbf{G} using only the subsets of \mathbf{U} , \mathbf{V} , and Λ defined in Equations (6.61), (6.66), and (6.67), we can rewrite the shifted eigenvalue problem given by Equations (6.53) and (6.54):

$$\begin{matrix} \mathbf{G} & \mathbf{V} & = & \mathbf{U} & \Lambda \\ N \times M & M \times M & & N \times N & N \times M \end{matrix} \quad (6.53)$$

and

$$\begin{matrix} \mathbf{G}^T & \mathbf{U} & = & \mathbf{V} & \Lambda^T \\ M \times N & N \times N & & M \times M & M \times N \end{matrix} \quad (6.54)$$

as

$$1. \quad \begin{matrix} \mathbf{G} & \mathbf{V}_P & = & \mathbf{U}_P & \Lambda_P \\ N \times M & M \times P & & N \times P & P \times P \end{matrix} \quad (6.62)$$

$$2. \quad \begin{matrix} \mathbf{G}^T & \mathbf{U}_P & = & \mathbf{V}_P & \Lambda_P \\ M \times N & N \times P & & M \times P & P \times P \end{matrix} \quad (6.63)$$

$$3. \quad \begin{matrix} \mathbf{G} & \mathbf{V}_0 & = & \mathbf{0} \\ N \times M & M \times (M - P) & & N \times (M - P) \end{matrix} \quad (6.64)$$

$$4. \quad \begin{matrix} \mathbf{G}^T & \mathbf{U}_0 & = & \mathbf{0} \\ M \times N & N \times (N - P) & & M \times (N - P) \end{matrix} \quad (6.65)$$

Note that the eigenvectors in \mathbf{V} are a set of M orthogonal vectors which span *model space*, while the eigenvectors in \mathbf{U} are a set of N orthogonal vectors which span *data space*. The P vectors in \mathbf{V}_P span a P -dimensional subset of model space, while the P vectors in \mathbf{U}_P span a P -dimensional subset of data space. \mathbf{V}_0 and \mathbf{U}_0 are called *null*, or *zero*, spaces. They are $(M - P)$ and $(N - P)$ dimensional subsets of model and data spaces, respectively.

6.8.3 Summarizing SVD

In summary, we started with Equations (1.13) and (6.5)

$$\mathbf{Gm} = \mathbf{d} \quad (1.13)$$

and

$$\mathbf{G}^T \mathbf{y} = \mathbf{c} \quad (6.5)$$

We constructed

$$\mathbf{B} = \begin{array}{c} \left[\begin{array}{c|c} \mathbf{0} & \mathbf{G} \\ \hline \mathbf{G}^T & \mathbf{0} \end{array} \right] \begin{array}{l} \uparrow \uparrow \\ N \\ \downarrow \downarrow \\ \uparrow \uparrow \\ M \\ \downarrow \downarrow \end{array} \\ \left| \leftarrow N \Rightarrow \right| \left| \leftarrow M \Rightarrow \right| \end{array} \quad (6.2)$$

We then considered the eigenvalue problem for \mathbf{B}

$$\mathbf{B} \mathbf{w}_i = \eta_i \mathbf{w}_i \quad i = 1, 2, \dots, (N + M) \quad (6.10)$$

This led us to the *shifted eigenvalue problem*

$$\mathbf{G} \mathbf{v}_i = \eta_i \mathbf{u}_i \quad i = 1, 2, \dots, (N + M) \quad (6.16)$$

and

$$\mathbf{G}^T \mathbf{u}_i = \eta_i \mathbf{v}_i \quad i = 1, 2, \dots, (N + M) \quad (6.17)$$

We found that the shifted eigenvalue problem leads us to eigenvalue problems for $\mathbf{G}^T \mathbf{G}$ and $\mathbf{G} \mathbf{G}^T$:

$$\mathbf{G}^T \mathbf{G} \mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad i = 1, 2, \dots, M \quad (6.20)$$

and

$$\mathbf{G} \mathbf{G}^T \mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad i = 1, 2, \dots, N \quad (6.28)$$

We then introduced the singular value λ_i , given by the positive square root of the eigenvalues from Equations (6.20) and (6.28)

$$\lambda_i = +\sqrt{\eta_i} \quad (6.49)$$

Equations (6.16), (6.17), (6.20) and (6.28) give us a way to find \mathbf{U} , \mathbf{V} , and Λ . They also lead, eventually, to

$$\mathbf{G} = \begin{array}{c} \mathbf{U} \quad \Lambda \quad \mathbf{V}^T \\ N \times M \quad N \times N \quad N \times M \quad M \times M \end{array} \quad (6.56)$$

We then considered partitioning the matrices based on P , the number of nonzero singular values. This led us to *singular-value decomposition*

$$\mathbf{G} = \begin{array}{c} \mathbf{U}_P \quad \Lambda_P \quad \mathbf{V}_P^T \\ N \times M \quad N \times P \quad P \times P \quad P \times M \end{array} \quad (6.65)$$

Before considering an inverse operator based on singular-value decomposition, it is probably useful to cover the mechanics of singular-value decomposition.

6.9 Mechanics of Singular-Value Decomposition

The steps involved in singular-value decomposition are as follows:

Step 1. Begin with $\mathbf{Gm} = \mathbf{d}$.

Form $\mathbf{G}^T\mathbf{G}$ ($M \times M$) or $\mathbf{G}\mathbf{G}^T$ ($N \times N$), whichever is smaller. (N.B. Typically, there are more observations than model parameters; thus, $N > M$, and $\mathbf{G}^T\mathbf{G}$ is the more common choice.)

Step 2. Solve the eigenvalue problem for Hermitian $\mathbf{G}^T\mathbf{G}$ (or $\mathbf{G}\mathbf{G}^T$)

$$\mathbf{G}^T\mathbf{G}\mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad (6.20)$$

1. Find the P nonzero η_i^2 and associated \mathbf{v}_i .
2. Let $\lambda_i = +(\eta_i^2)^{1/2}$.
3. Form

$$\mathbf{V}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P \\ \vdots & \vdots & \cdots & \vdots \\ & & & \vdots \\ & & & M \times P \end{bmatrix} \quad (6.67)$$

and

$$\Lambda_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_P \\ & & & P \times P \end{bmatrix} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_P \quad (6.61)$$

Step 3. Use $\mathbf{G}\mathbf{v}_i = \lambda_i\mathbf{u}_i$ to find \mathbf{u}_i for each known λ_i, \mathbf{v}_i .

Note: Finding \mathbf{u}_i this way preserves the *sign* relationship implicit between $\mathbf{u}_i, \mathbf{v}_i$ by taking the positive member of each pair $(+\lambda_i, -\lambda_i)$. You will *not* preserve the sign relationship (except by luck) if you use $\mathbf{G}\mathbf{G}^T\mathbf{u}_i = \eta_i^2 \mathbf{u}_i$ to find \mathbf{u}_i .

Step 4. Form

$$\mathbf{U}_P = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_P \\ \vdots & \vdots & \cdots & \vdots \\ & & & \vdots \\ & & & N \times P \end{bmatrix} \quad (6.66)$$

Step 5. Finally, form \mathbf{G} as

$$\mathbf{G} = \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \quad (6.65)$$

$$\begin{matrix} N \times M & N \times P & P \times P & P \times M \end{matrix}$$

6.10 Implications of Singular-Value Decomposition

6.10.1 Relationships Between \mathbf{U} , \mathbf{U}_P , and \mathbf{U}_0

1. $\mathbf{U}^T \mathbf{U} = \mathbf{U} \mathbf{U}^T = \mathbf{I}_N$ Since \mathbf{U} is an orthogonal matrix.
 $\begin{matrix} N \times N & N \times N & N \times N & N \times N & N \times N \end{matrix}$
2. $\mathbf{U}_P^T \mathbf{U}_P = \mathbf{I}_P$ \mathbf{U}_P is semiorthogonal because all P vectors in \mathbf{U}_P are perpendicular to each other.
 $\begin{matrix} P \times N & N \times P \end{matrix}$
3. $\mathbf{U}_P \mathbf{U}_P^T \neq \mathbf{I}_N$ (Unless $P = N$.) $\mathbf{U}_P \mathbf{U}_P^T$ is $N \times N$. \mathbf{U}_P cannot span N -space with only P (N -dimensional) vectors.
 $\begin{matrix} N \times P & P \times N \end{matrix}$
4. $\mathbf{U}_0^T \mathbf{U}_0 = \mathbf{I}_{N-P}$ \mathbf{U}_0 is semiorthogonal since the $(N - P)$ vectors in \mathbf{U}_0 are all perpendicular to each other.
 $\begin{matrix} (N - P) \times N & N \times (N - P) \end{matrix}$
5. $\mathbf{U}_0 \mathbf{U}_0^T \neq \mathbf{I}_N$ \mathbf{U}_0 has $(N - P)$ N -dimensional vectors in it. It cannot span N -space. $\mathbf{U}_0 \mathbf{U}_0^T$ is $N \times N$.
 $\begin{matrix} N \times (N - P) & (N - P) \times N \end{matrix}$
6. $\mathbf{U}_P^T \mathbf{U}_0 = \mathbf{0}$ Since all the eigenvectors in \mathbf{U}_P are perpendicular to all the eigenvectors in \mathbf{U}_0 .
 $\begin{matrix} P \times N & N \times (N - P) & P \times (N - P) \end{matrix}$
7. $\mathbf{U}_0^T \mathbf{U}_P = \mathbf{0}$ Again, since all the eigenvectors in \mathbf{U}_0 are perpendicular to all the eigenvectors in \mathbf{U}_P .
 $\begin{matrix} (N - P) \times N & N \times P & (N - P) \times P \end{matrix}$

6.10.2 Relationships Between \mathbf{V} , \mathbf{V}_P , and \mathbf{V}_0

1. $\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}_M$ Since \mathbf{V} is an orthogonal matrix.
 $\begin{matrix} M \times M & M \times M & M \times M & M \times M \end{matrix}$
2. $\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_P$ \mathbf{V}_P is semiorthogonal because all P vectors in \mathbf{V}_P are perpendicular to each other.
 $\begin{matrix} P \times M & M \times P \end{matrix}$
3. $\mathbf{V}_P \mathbf{V}_P^T \neq \mathbf{I}_M$ (Unless $P = M$.) \mathbf{V}_P^T is $M \times M$. \mathbf{V}_P cannot span M -space with only P (M -dimensional) vectors.
 $\begin{matrix} M \times P & P \times M \end{matrix}$

4.
$$\begin{matrix} \mathbf{V}_0^T & \mathbf{V}_0 & = & \mathbf{I}_{M-P} \\ (M-P) \times M & M \times (M-P) & & \end{matrix}$$
 \mathbf{V}_0 is semiorthogonal since the $(M - P)$ vectors in \mathbf{V}_0 are all perpendicular to each other.
5.
$$\begin{matrix} \mathbf{V}_0 & \mathbf{V}_0^T & \neq & \mathbf{I}_M \\ M \times (M-P) & (M-P) \times M & & \end{matrix}$$
 \mathbf{V}_0 has $(M - P)$ M -dimensional vectors in it. It cannot span M -space. $\mathbf{V}_0 \mathbf{V}_0^T$ is $M \times M$.
6.
$$\begin{matrix} \mathbf{V}_P^T & \mathbf{V}_0 & = & \mathbf{0} \\ P \times M & M \times (M-P) & & P \times (M-P) \end{matrix}$$
 Since all the eigenvectors in \mathbf{V}_P are perpendicular to all the eigenvectors in \mathbf{V}_0 .
7.
$$\begin{matrix} \mathbf{V}_0^T & \mathbf{V}_P & = & \mathbf{0} \\ (M-P) \times M & M \times P & & (M-P) \times P \end{matrix}$$
 Again, since all the eigenvectors in \mathbf{V}_0 are perpendicular to all the eigenvectors in \mathbf{V}_P .

6.10.3 Graphic Representation of \mathbf{U} , \mathbf{U}_P , \mathbf{U}_0 , \mathbf{V} , \mathbf{V}_P , \mathbf{V}_0 Spaces

Recall that starting with Equation (1.13)

$$\begin{matrix} \mathbf{G} & \mathbf{m} & = & \mathbf{d} \\ N \times M & M \times 1 & & N \times 1 \end{matrix} \quad (1.13)$$

we obtained the fundamental decomposition theorem

$$\begin{matrix} \mathbf{G} & = & \mathbf{U} & \Lambda & \mathbf{V}^T \\ N \times M & & N \times N & N \times M & M \times M \end{matrix} \quad (6.56)$$

and singular-value decomposition

$$\begin{matrix} \mathbf{G} & = & \mathbf{U}_P & \Lambda_P & \mathbf{V}_P^T \\ N \times M & & N \times P & P \times P & P \times M \end{matrix} \quad (6.65)$$

This gives us the following:

1. Recall the definitions of \mathbf{U} , \mathbf{U}_P , and \mathbf{U}_0

$$\begin{matrix} \mathbf{U} \\ N \times N \end{matrix} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_P & \mathbf{u}_{P+1} & \cdots & \mathbf{u}_N \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} = [\mathbf{U}_P \mid \mathbf{U}_0] \quad (6.58)$$

2. Similarly, recall the definitions for \mathbf{V} , \mathbf{V}_P and \mathbf{V}_0

$$\begin{matrix} \mathbf{V} \\ M \times M \end{matrix} = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \cdots & \mathbf{v}_P & \mathbf{v}_{P+1} & \cdots & \mathbf{v}_N \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix} = [\mathbf{V}_P \mid \mathbf{V}_0] \quad (6.59)$$

3. Combining \mathbf{U} , \mathbf{U}_P , \mathbf{U}_0 , and \mathbf{V} , \mathbf{V}_P , \mathbf{V}_0 graphically

$$\begin{array}{c}
 | \leftarrow P \Rightarrow | | \leftarrow N - P \Rightarrow | \\
 \updownarrow \left[\begin{array}{c|c}
 \mathbf{U}_P & \mathbf{U}_0 \\
 \hline
 \mathbf{V}_P & \mathbf{V}_0
 \end{array} \right] \\
 \updownarrow \\
 | \leftarrow P \Rightarrow | | \leftarrow M - P \Rightarrow |
 \end{array} \quad (6.68)$$

4. Summarizing:

- (1) \mathbf{V} is an $M \times M$ matrix with the eigenvectors of $\mathbf{G}^T \mathbf{G}$ as columns. It is an orthogonal matrix.
- (2) \mathbf{V}_P is an $M \times P$ matrix with the P eigenvectors of $\mathbf{G}^T \mathbf{G}$ associated with nonzero eigenvalues of $\mathbf{G}^T \mathbf{G}$. \mathbf{V}_P is a semiorthogonal matrix.
- (3) \mathbf{V}_0 is an $M \times (M - P)$ matrix with the $M - P$ eigenvectors of $\mathbf{G}^T \mathbf{G}$ associated with the zero eigenvalues of $\mathbf{G}^T \mathbf{G}$. \mathbf{V}_0 is a semiorthogonal matrix.
- (4) The eigenvectors in \mathbf{V} , \mathbf{V}_P , or \mathbf{V}_0 are all M -dimensional vectors. They are all associated with *model* space, since \mathbf{m} , the model parameter vector of $\mathbf{G}\mathbf{m} = \mathbf{d}$, is an M -dimensional vector.
- (5) \mathbf{U} is an $N \times N$ matrix with the eigenvectors of $\mathbf{G}\mathbf{G}^T$ as columns. It is an orthogonal matrix.
- (6) \mathbf{U}_P is an $N \times P$ matrix with the P eigenvectors of $\mathbf{G}\mathbf{G}^T$ associated with the nonzero eigenvalues of $\mathbf{G}\mathbf{G}^T$. \mathbf{U}_P is a semiorthogonal matrix.
- (7) \mathbf{U}_0 is an $N \times (N - P)$ matrix with the $N - P$ eigenvectors of $\mathbf{G}\mathbf{G}^T$ associated with the zero eigenvalues of $\mathbf{G}\mathbf{G}^T$. \mathbf{U}_0 is a semiorthogonal matrix.
- (8) The eigenvectors of \mathbf{U} , \mathbf{U}_P or \mathbf{U}_0 are all N -dimensional vectors. They are all associated with *data* space, since \mathbf{d} , the data vector of $\mathbf{G}\mathbf{m} = \mathbf{d}$, is an N -dimensional vector.

6.11 Classification of $\mathbf{d} = \mathbf{G}\mathbf{m}$ Based on P , M , and N

6.11.1 Introduction

In Section 3.3 we introduced a classification of the system of equations

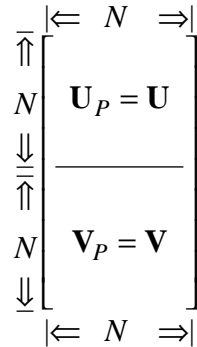
$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

based on the dimensions of \mathbf{d} ($N \times 1$) and \mathbf{m} ($M \times 1$). I said at the time that I found the classification lacking, and would return to it later. Now that we have considered singular-value decomposition, including finding P , the number of nonzero singular values, I would like to introduce a better classification scheme.

There are four basic classes of problems, based on the relationship between P , M , and N . We will consider each class one at a time below.

6.11.2 Class I: $P = M = N$

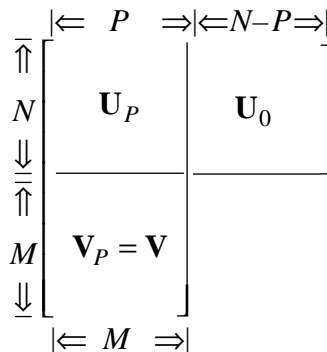
Graphically, for this case, we have



1. \mathbf{U}_0 and \mathbf{V}_0 are empty.
2. \mathbf{G} has a unique, mathematical inverse \mathbf{G}^{-1} .
3. There is a unique solution for \mathbf{m} .
4. The data can be fit exactly.

6.11.3 Class II: $P = M < N$

Graphically, for this case, we have



1. \mathbf{V}_0 is empty since $P = M$.
2. \mathbf{U}_0 is not empty since $P < N$.
3. \mathbf{G} has no mathematical inverse.
4. There is a unique solution in the sense that only one solution has the smallest misfit to the data.
5. The data cannot be fit exactly unless the *compatibility equations* are satisfied, which are defined as follows:

$$\begin{matrix} \mathbf{U}_0^T & \mathbf{d} & = & \mathbf{0} \\ (N-P) \times N & N \times 1 & & (N-P) \times 1 \end{matrix} \quad (6.69)$$

If the compatibility equations are satisfied, one can fit the data exactly. The compatibility equations are equivalent to saying that \mathbf{d} has no projection onto \mathbf{U}_0 .

Equation (6.69) can be thought of as the $N - P$ dot products of the eigenvectors in \mathbf{U}_0 with the data vector \mathbf{d} . If all of the dot products are zero, then \mathbf{d} has no component in the $(N - P)$ -dimensional subset of N -space spanned by the vectors in \mathbf{U}_0 . \mathbf{G} , operating on any vector \mathbf{m} , can only predict a vector that lies in the P -dimensional subset of N -space spanned by the P eigenvectors in \mathbf{U}_P . We will return to this later.

6. $P = M < N$ is the classic least squares environment. We will consider least squares again when we introduce the generalized inverse.

6.11.4 Class III: $P = N < M$

Graphically, for this case, we have



1. \mathbf{U}_0 is empty since $P = N$.
2. \mathbf{V}_0 is not empty since $P < M$.
3. \mathbf{G} has no mathematical inverse.

4. You can fit the data exactly because \mathbf{U}_0 is empty.
5. Solution is not unique. If \mathbf{m}^{est} is a solution which fits the data exactly

$$\mathbf{G}\mathbf{m}^{\text{est}} = \mathbf{d}^{\text{pre}} = \mathbf{d} \quad (6.70)$$

then

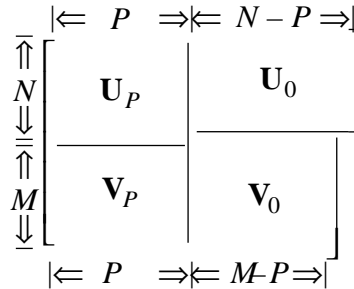
$$\mathbf{m}^{\text{est}} + \sum_{i=P+1}^M \alpha_i \mathbf{v}_i \quad (6.71)$$

is also a solution, where α_i is any arbitrary constant.

6. $P = M < N$ is the minimum length environment. The minimum length solution sets $\alpha_i, i = (P + 1), \dots, M$ to zero.

6.11.5 Class IV: $P < \min(N, M)$

Graphically, for this case, we have



1. Neither \mathbf{U}_0 nor \mathbf{V}_0 is empty.
2. \mathbf{G} has no mathematical inverse.
3. You cannot fit the data exactly unless the compatibility equations (Equation 6.69) are satisfied.
4. The solution is nonunique.

This sounds like a pretty bleak environment. No mathematical inverse. Cannot fit the data. The solution is nonunique. It probably comes as no surprise that most realistic problems are of this type [$P < \min(N, M)$]!

In the next chapter we will introduce the *generalized inverse operator*. It will reduce to the unique mathematical inverse when $P = M = N$. It will reduce to the least squares operator when we have $P = M < N$, and to the minimum length operator when $P = N < M$. It will also give us a solution in the general case where we have $P < \min(N, M)$ that has many of the properties of the least squares and minimum length solutions.

CHAPTER 7: THE GENERALIZED INVERSE AND MEASURES OF QUALITY

7.1 Introduction

Thus far we have used the shifted eigenvalue problem to do singular-value decomposition for the system of equations $\mathbf{G}\mathbf{m} = \mathbf{d}$. That is, we have

$$\begin{array}{ccccc} \mathbf{G} & = & \mathbf{U} & \Lambda & \mathbf{V}^T \\ N \times M & & N \times N & N \times M & M \times M \end{array} \quad (6.56)$$

and also

$$\begin{array}{ccccc} \mathbf{G} & = & \mathbf{U}_P & \Lambda_P & \mathbf{V}_P^T \\ N \times M & & N \times P & P \times P & P \times M \end{array} \quad (6.65)$$

where \mathbf{U} is an $N \times N$ orthogonal matrix, and where the i th column is given by the i th eigenvector \mathbf{u}_i which satisfies

$$\mathbf{G}\mathbf{G}^T\mathbf{u}_i = \eta_i^2 \mathbf{u}_i \quad (6.28)$$

\mathbf{V} is an $M \times M$ orthogonal matrix, where the i th column is given by the i th eigenvector \mathbf{v}_i which satisfies

$$\mathbf{G}^T\mathbf{G}\mathbf{v}_i = \eta_i^2 \mathbf{v}_i. \quad (6.20)$$

Λ is an $N \times M$ diagonal matrix with the singular values $\lambda_i = \sqrt{\eta_i^2}$ along the diagonal. \mathbf{U}_P , Λ_P , and \mathbf{V}_P are the subsets of \mathbf{U} , Λ , and \mathbf{V} , respectively, associated with the P nonzero singular values, $P \leq \min(N, M)$.

We found four classes of problems for $\mathbf{G}\mathbf{m} = \mathbf{d}$ based on P , N , M :

Class I: $P = N = M$; \mathbf{G}^{-1} (mathematical) exists.

Class II: $P = M < N$; least squares. Recall $\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d}$.

Class III: $P = N < M$; Minimum Length. Recall $\mathbf{m}_{ML} = \langle \mathbf{m} \rangle + \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} \times [\mathbf{d} - \mathbf{G}\langle \mathbf{m} \rangle]$.

Class IV: $P < \min(N, M)$; at present, we have no way of obtaining an \mathbf{m}^{est} .

Thus, in this chapter we seek an inverse operator that has the following properties:

1. Reduces to \mathbf{G}^{-1} when $P = N = M$.
2. Reduces to $[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T$ when $P = M < N$ (least squares).
3. Reduces to $\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}$ when $P = N < M$ (minimum length).
4. Exists when $P < \min(N, M)$.

In the following pages we will consider each of these classes of problems, beginning with $P = N = M$ (Class I). In this case,

$$\mathbf{G} = \mathbf{U} \quad \Lambda \quad \mathbf{V}^T \quad (7.1)$$

$N \times N \quad N \times N \quad N \times N \quad N \times N$

with

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{bmatrix} \quad (7.2)$$

Since $P = N = M$, there are no zero singular values and we have

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0 \quad (7.3)$$

In order to find an inverse operator based on Equation (7.1), we need to find the inverse of a product of matrices. Applying the results from Equation (2.8) to Equation (7.1) above gives

$$\mathbf{G}^{-1} = [\mathbf{V}^T]^{-1}\Lambda^{-1} \mathbf{U}^{-1} \quad (7.4)$$

We know Λ^{-1} exists and is given by

$$\Lambda^{-1} = \begin{bmatrix} 1/\lambda_1 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_N \end{bmatrix} \quad (7.5)$$

$N \times N$

We now make use of the fact that both \mathbf{U} and \mathbf{V} are orthogonal matrices. The properties of \mathbf{U} and \mathbf{V} that we wish to use are

$$[\mathbf{V}^T]^{-1} = \mathbf{V} \quad (7.6)$$

and

$$\mathbf{U}^{-1} = \mathbf{U}^T \quad (7.7)$$

Therefore

$$\boxed{\begin{array}{cccc} \mathbf{G}^{-1} = & \mathbf{V} & \Lambda^{-1} & \mathbf{U}^T & P = N = M \\ N \times N & N \times N & N \times N & N \times N & \end{array}} \quad (7.8)$$

Equation (7.8) implies that \mathbf{G}^{-1} , the mathematical inverse of \mathbf{G} , can be found using singular-value decomposition when $P = N = M$.

What we need now is to find an operator for the other three classes of problems that will reduce to the mathematical inverse \mathbf{G}^{-1} when it exists.

7.2 The Generalized Inverse Operator \mathbf{G}_g^{-1}

7.2.1 Background Information

We start out with three pieces of information:

$$1. \quad \mathbf{G} = \mathbf{U}\Lambda\mathbf{V}^T \quad (6.70)$$

$$2. \quad \mathbf{G}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{U}^T \text{ (when } \mathbf{G}^{-1} \text{ exists)} \quad (7.8)$$

$$3. \quad \mathbf{G} = \mathbf{U}_P\Lambda_P\mathbf{V}_P^T \text{ (singular-value decomposition)} \quad (6.65)$$

Then, by analogy with defining the inverse in Equation (7.8) above on the form of Equation (7.1), we introduce the *generalized inverse operator*:

$$\boxed{\begin{array}{cccc} \mathbf{G}_g^{-1} = & \mathbf{V}_P & \Lambda_P^{-1} & \mathbf{U}_P^T \\ M \times N & M \times P & P \times P & P \times N \end{array}} \quad (7.9)$$

and find out the consequences for our four cases. Menke points out that there may be many generalized inverses, but Equation (7.9) is by far the most common generalized inverse.

7.2.2 Class I: $P = N = M$

In this case, we have

$$1. \quad \mathbf{V}_P = \mathbf{V} \text{ and } \mathbf{U}_P = \mathbf{U}.$$

$$2. \quad \mathbf{V}_0 \text{ and } \mathbf{U}_0 \text{ are empty.}$$

We start with the definition of the generalized inverse operator in Equation (7.9):

$$\mathbf{G}_g^{-1} = \mathbf{V}_P\Lambda_P^{-1}\mathbf{U}_P^T \quad (7.9)$$

But, since $P = M$ we have

$$\mathbf{V}_P = \mathbf{V} \quad (7.10)$$

Similarly, since $P = N$ we have

$$\mathbf{U}_P = \mathbf{U} \quad (7.11)$$

Finally, since $P = N = M$, we have

$$\Lambda_P^{-1} = \Lambda^{-1} \quad (7.12)$$

Therefore, combining Equations (7.9)–(7.12), we recover Equation (7.8)

$$\mathbf{G}^{-1} = \mathbf{V}\Lambda^{-1}\mathbf{U}^T \quad P = N = M \quad (7.8)$$

the exact mathematical inverse. Thus, we have shown that the generalized inverse operator reduces to the exact mathematical inverse in the case of $P = N = M$. Next we consider the case of $P = M < N$.

7.2.3 Class II: $P = M < N$

This is the least squares environment where we have more observations than unknowns, but where a unique solution exists. In this case, we have:

1. $\mathbf{V}_P = \mathbf{V}$
2. \mathbf{V}_0 is empty.
3. \mathbf{U}_0 exists.

Ultimately we wish to show that the generalized inverse operator reduces to the least squares operator when $P = M < N$.

The Role of $\mathbf{G}^T\mathbf{G}$

Recall that the least squares operator, as defined in Equation (3.27), for example, is given by

$$\mathbf{m}_{LS} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d} \quad (3.27)$$

We first consider $\mathbf{G}^T\mathbf{G}$, using singular-value decomposition for \mathbf{G} , obtaining

$$\mathbf{G}^T\mathbf{G} = [\mathbf{U}_P\Lambda_P\mathbf{V}_P^T]^T[\mathbf{U}_P\Lambda_P\mathbf{V}_P^T] \quad (7.13)$$

Recall that the transpose of the product of matrices is given by the product of the transposes in the reverse order:

$$[\mathbf{AB}]^T = \mathbf{B}^T\mathbf{A}^T$$

Therefore

$$\mathbf{G}^T\mathbf{G} = [\mathbf{V}_P^T]^T \Lambda_P^T \mathbf{U}_P^T \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \quad (7.14)$$

or

$$\mathbf{G}^T\mathbf{G} = \mathbf{V}_P \Lambda_P \mathbf{U}_P^T \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \quad (7.15)$$

since

$$\Lambda_P^T = \Lambda_P \quad (7.16)$$

and

$$[\mathbf{V}_P^T]^T = \mathbf{V}_P \quad (7.17)$$

We know, however, that \mathbf{U}_P is a semiorthogonal matrix. Thus

$$\mathbf{U}_P^T \mathbf{U}_P = \mathbf{I}_P \quad (7.18)$$

Therefore, Equation (7.15) reduces to

$$\mathbf{G}^T\mathbf{G} = \mathbf{V}_P \Lambda_P \Lambda_P \mathbf{V}_P^T \quad (7.19)$$

Now, consider the product of Λ_P with itself

$$\Lambda_P \Lambda_P = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p \end{bmatrix} \quad (7.20)$$

or

$$\Lambda_P \Lambda_P = \begin{bmatrix} \lambda_1^2 & 0 & \cdots & 0 \\ 0 & \lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_p^2 \end{bmatrix} = \Lambda_P^2 \quad (7.21)$$

where we have introduced the notation Λ_P^2 for the product of Λ_P with itself.

Please be aware, as noted before, that the notation \mathbf{A}^2 , when \mathbf{A} is a matrix, has no universally accepted definition. We will use the definition implied by Equation (7.21). Finally, then, we write Equation (7.19) as

$$\begin{matrix} \mathbf{G}^T\mathbf{G} & = & \mathbf{V}_P & \Lambda_P^2 & \mathbf{V}_P^T \\ M \times M & & M \times P & P \times P & P \times M \end{matrix} \quad (7.22)$$

Finding the Inverse of $\mathbf{G}^T\mathbf{G}$

I claim that $\mathbf{G}^T\mathbf{G}$ has a mathematical inverse $[\mathbf{G}^T\mathbf{G}]^{-1}$ when $P = M$. The reason that $[\mathbf{G}^T\mathbf{G}]^{-1}$ exists in this case is that $\mathbf{G}^T\mathbf{G}$ has the following eigenvalue problem:

$$\mathbf{G}^T\mathbf{G}\mathbf{v}_i = \eta_i^2 \mathbf{v}_i \quad i = 1, \dots, M \quad (6.20)$$

where, because $P = M$, we know that all M η_i^2 are nonzero. That is, $\mathbf{G}^T\mathbf{G}$ has no zero eigenvalues. Thus, it has a mathematical inverse.

Using the theorem presented earlier about the inverse of a product of matrices in Equations (2.8), we have

$$[\mathbf{G}^T\mathbf{G}]^{-1} = [\mathbf{V}_P^T]^{-1}[\Lambda_P^2]^{-1}\mathbf{V}_P^{-1} \quad (7.23)$$

The inverse of \mathbf{V}_P^T is found as follows. First,

$$\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_P \quad (7.24)$$

is always true because \mathbf{V}_P is semiorthogonal. But, because we have that $P = M$ in this case, we also have

$$\mathbf{V}_P \mathbf{V}_P^T = \mathbf{I}_M \quad (7.25)$$

Thus, \mathbf{V}_P is itself an orthogonal matrix, and we have

$$[\mathbf{V}_P^T]^{-1} = \mathbf{V}_P \quad (7.26)$$

and

$$\mathbf{V}_P^{-1} = \mathbf{V}_P^T \quad (7.27)$$

Thus, we can write Equation (7.23) as

$$[\mathbf{G}^T\mathbf{G}]^{-1} = \mathbf{V}_P[\Lambda_P^2]^{-1}\mathbf{V}_P^T \quad (7.28)$$

Finally, we note that $[\Lambda_P^2]^{-1}$ is given by

$$[\Lambda_P^2]^{-1} = \Lambda_P^{-2} = \begin{bmatrix} 1/\lambda_1^2 & 0 & \dots & 0 \\ 0 & 1/\lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & 1/\lambda_P^2 \end{bmatrix} \quad (7.29)$$

Therefore

$$[\mathbf{G}^T\mathbf{G}]^{-1} = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \quad (7.30)$$

where Λ_P^{-2} is as defined in Equation (7.29).

Equivalence of \mathbf{G}_g^{-1} and Least Squares When $P = M < N$

We start with the least squares operator, from Equation (3.28), for example

$$\mathbf{G}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.28)$$

We can use Equation (7.30) for $[\mathbf{G}^T \mathbf{G}]^{-1}$ in Equation (3.28) and singular-value decomposition for \mathbf{G}^T to obtain

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T [\mathbf{U}_P \Lambda_P \mathbf{V}_P^T]^T \quad (7.31)$$

$$= \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \mathbf{V}_P \Lambda_P \mathbf{U}_P^T \quad (7.32)$$

But, because \mathbf{V}_P is semiorthogonal, we have from Equation (7.24)

$$\mathbf{V}_P^T \mathbf{V}_P = \mathbf{I}_P \quad (7.24)$$

Thus, Equation (7.32) becomes

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-2} \Lambda_P \mathbf{U}_P^T \quad (7.33)$$

Now, considering Equations (7.20) and (7.21), we see that

$$\Lambda_P^{-2} \Lambda_P = \Lambda_P^{-1} \quad (7.34)$$

Finally, then, Equation (7.33) becomes

$$[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T = \mathbf{G}_g^{-1} \quad (7.35)$$

as required. That is, we have shown that when $P = M < N$, the generalized inverse operator is equivalent to the least squares operator.

Geometrical Interpretation of \mathbf{G}_g^{-1} When $P = M < N$

It is possible to gain some insight into the generalized inverse operator by considering a geometrical argument. An arbitrary data vector \mathbf{d} may have components in both \mathbf{U}_P and \mathbf{U}_0 spaces. The generalized inverse operator returns a solution \mathbf{m}_g , for which the predicted data lies completely in \mathbf{U}_P space and minimizes the misfit to the observed data. The steps necessary to see this follow:

Step 1. Let \mathbf{m}_g be the generalized inverse solution, given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \quad (7.36)$$

Step 2. Let $\hat{\mathbf{d}}$ be the predicted data, given by

$$\hat{\mathbf{d}} = \mathbf{G}\mathbf{m}_g \quad (7.37)$$

We now introduce the following theorem about the relationship of the predicted data to \mathbf{U}_P space

Theorem: $\hat{\mathbf{d}}$ lies completely in \mathbf{U}_P -space (the subset of N -space spanned by the P eigenvectors in \mathbf{U}_P).

Proof: If $\hat{\mathbf{d}}$ lies in \mathbf{U}_P -space, it is orthogonal to \mathbf{U}_0 -space. That is,

$$\begin{aligned} \mathbf{U}_0^T \hat{\mathbf{d}} &= \mathbf{U}_0^T \mathbf{G}\mathbf{m}_g = \mathbf{U}_0^T \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{m}_g \\ &= \mathbf{0} \\ &\quad (N-P) \times 1 \end{aligned} \quad (7.38)$$

which follows from

$$\mathbf{U}_0^T \mathbf{U}_P = \mathbf{0} \quad (7.39)$$

That is, every eigenvector in \mathbf{U}_0 is perpendicular to every eigenvector in \mathbf{U}_P . Another way to see this is that *all* of the eigenvectors in \mathbf{U} are perpendicular to each other. Thus, *any* subset of \mathbf{U} is perpendicular to the rest of \mathbf{U} . Q.E.D.

Step 3. Let $\mathbf{d} - \hat{\mathbf{d}}$ be the residual data vector (i.e., observed – predicted data, also known as the misfit vector), given by

$$\begin{aligned} \mathbf{d} - \hat{\mathbf{d}} &= \mathbf{d} - \mathbf{G}\mathbf{m}_g \\ &= \mathbf{d} - \mathbf{G}[\mathbf{G}_g^{-1} \mathbf{d}] \\ &= \mathbf{d} - \mathbf{G}\mathbf{G}_g^{-1} \mathbf{d} \\ &= \mathbf{d} - \{\mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T\} \{\mathbf{V}_P \mathbf{\Lambda}_P \mathbf{U}_P^T\} \mathbf{d} \\ &\quad \vdots \\ &= \mathbf{d} - \mathbf{U}_P \mathbf{U}_P^T \mathbf{d} \end{aligned} \quad (7.40)$$

We cannot further reduce Equation (7.40) whenever $P < N$ because in this case

$$\mathbf{U}_P \mathbf{U}_P^T \neq \mathbf{I}_N$$

Next, we introduce a theorem about the relationship between the misfit vector and \mathbf{U}_0 space.

Theorem: The misfit vector $\mathbf{d} - \hat{\mathbf{d}}$ is orthogonal to \mathbf{U}_P .

Proof:

$$\begin{aligned} \mathbf{U}_P^T [\mathbf{d} - \hat{\mathbf{d}}] &= \mathbf{U}_P^T [\mathbf{d} - \mathbf{U}_P \mathbf{U}_P^T \mathbf{d}] \\ &= \mathbf{U}_P^T \mathbf{d} - \mathbf{U}_P^T \mathbf{U}_P \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{U}_P^T \mathbf{d} - \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{0} \quad \text{Q.E.D.} \end{aligned} \tag{7.41}$$

$P \times 1$

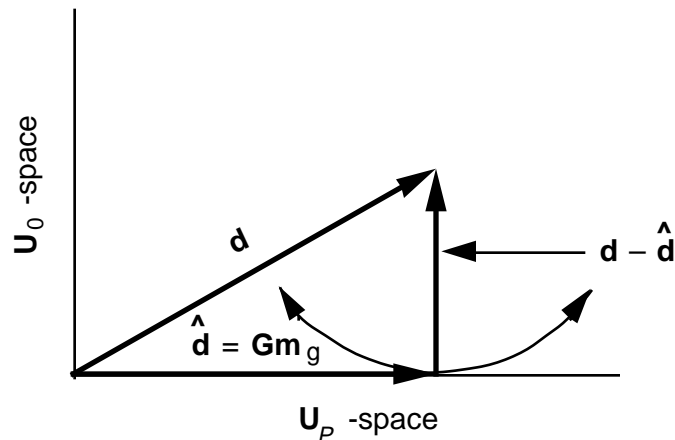
The crucial step in going from the second to third lines being that $\mathbf{U}_P^T \mathbf{U}_P = \mathbf{I}_P$ since \mathbf{U}_P is semiorthogonal. This implies that the misfit vector $\mathbf{d} - \hat{\mathbf{d}}$ lies completely in the space spanned by \mathbf{U}_0 .

Combining the results from the above two theorems, we introduce the final theorem of this section concerning the relationship between the predicted data and the misfit vector.

Theorem: The predicted data vector $\hat{\mathbf{d}}$ is perpendicular to the misfit vector $\mathbf{d} - \hat{\mathbf{d}}$.

- Proof:**
1. The predicted data vector $\hat{\mathbf{d}}$ lies in \mathbf{U}_P space.
 2. The misfit vector $\mathbf{d} - \hat{\mathbf{d}}$ lies in \mathbf{U}_0 space.
 3. Since the vectors in \mathbf{U}_P are perpendicular to the vectors in \mathbf{U}_0 , $\hat{\mathbf{d}}$ is perpendicular to the misfit vector $\mathbf{d} - \hat{\mathbf{d}}$. Q.E.D.

Step 4. Consider the following schematic graph showing the relationship between the various vectors and spaces:



The data vector \mathbf{d} has components in both \mathbf{U}_P and \mathbf{U}_0 spaces. Note the following points:

1. The predicted data vector $\hat{\mathbf{d}} = \mathbf{G}\mathbf{m}_g$ lies completely in \mathbf{U}_P space.
2. The residual vector $\mathbf{d} - \hat{\mathbf{d}}$ lies completely in \mathbf{U}_0 space.
3. The shortest distance from the observed data \mathbf{d} to the \mathbf{U}_P axis is given by the misfit vector $\mathbf{d} - \hat{\mathbf{d}}$.

Thus the generalized inverse \mathbf{G}_g^{-1} minimizes the distance between the observed data vector \mathbf{d} and \mathbf{U}_P , the subset of data space in which all possible predicted data $\hat{\mathbf{d}}$ must lie.

Recall that the least squares operator given in Equation (3.28)

$$\mathbf{G}_{LS}^{-1} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T \quad (3.28)$$

minimizes the length of the misfit vector. Thus, the generalized inverse operator is equivalent to the least squares operator when $P = M < N$.

Step 5. For $P = M < N$, it is possible to write the generalized inverse without forming \mathbf{U}_P . To see this, note that the generalized inverse is equivalent to least squares for $P = M < N$. That is,

$$\mathbf{G}_g^{-1} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T \quad (7.42)$$

But, by Equation (7.28), $[\mathbf{G}^T\mathbf{G}]^{-1}$ is given by

$$[\mathbf{G}^T\mathbf{G}]^{-1} = \mathbf{V}_P\mathbf{\Lambda}_P^{-2}\mathbf{V}_P^T \quad (7.28)$$

Thus, the generalized inverse in this case is given by

$$\mathbf{G}_g^{-1} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T = \mathbf{V}_P\mathbf{\Lambda}_P^{-2}\mathbf{V}_P^T\mathbf{G}^T \quad (7.43)$$

Equation (7.43) shows that the generalized inverse can be found without ever forming \mathbf{U}_P when $P = M < N$. In general, this shortcut is not used, even though you can form the inverse, because there is useful information lost about data space.

Step 6. Finally, recall the compatibility equations given by

$$\mathbf{U}_0^T \mathbf{d} = \mathbf{0} \quad (6.69)$$

$(N - P) \times 1$

Note that if the observed data \mathbf{d} has any projection in \mathbf{U}_0 space, is not possible to find a solution \mathbf{m} that can fit the data exactly. All estimates \mathbf{m} lead to predicted data $\mathbf{G}\mathbf{m}$ that lie in \mathbf{U}_P space. Thus, from the graph above, one sees that if the observed data, \mathbf{d} , lies completely in \mathbf{U}_P space, the compatibility equations are automatically satisfied.

7.2.4 Class III: $P = N < M$

This is the minimum length environment where we have more model parameters than observations. There are an infinite number of possible solutions that can fit the data exactly. Recall that the minimum length solution is the one which has the shortest length. Ultimately we wish to show that the generalized inverse operator reduces to the minimum length operator when $P = N < M$.

For $P = N < M$ we have

1. $\mathbf{U}_P = \mathbf{U}$
2. \mathbf{U}_0 is empty.
3. \mathbf{V}_0 is not empty.

The Role of $\mathbf{G}\mathbf{G}^T$

Recall that the minimum length operator, as defined in Equation (3.59), is given by

$$\mathbf{G}_{\text{ML}}^{-1} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.59)$$

We seek, thus, to show that $\mathbf{G}_g^{-1} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}$ in this case. First consider writing $\mathbf{G}\mathbf{G}^T$ using singular-value decomposition:

$$\begin{aligned} \mathbf{G}\mathbf{G}^T &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T [\mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T]^T \\ &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{V}_P \mathbf{\Lambda}_P \mathbf{U}_P^T \\ &= \mathbf{U}_P \mathbf{\Lambda}_P^2 \mathbf{U}_P^T \end{aligned} \quad (7.44)$$

Finding the Inverse of $\mathbf{G}\mathbf{G}^T$

Note that $\mathbf{G}\mathbf{G}^T$ is $N \times N$ and $P = N$. This implies that $[\mathbf{G}\mathbf{G}^T]^{-1}$, the mathematical inverse of $\mathbf{G}\mathbf{G}^T$, exists. Again using the theorem stated in Equation (2.8) about the inverse of a product of matrices, we have

$$\begin{aligned} [\mathbf{G}\mathbf{G}^T]^{-1} &= [\mathbf{U}_P^T]^{-1} [\mathbf{\Lambda}_P^2]^{-1} \mathbf{U}_P^{-1} \\ &= \mathbf{U}_P \mathbf{\Lambda}_P^{-2} \mathbf{U}_P^T \quad P = N \end{aligned} \quad (7.45)$$

Then

$$\mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} = [\mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T]^T \mathbf{U}_P \mathbf{\Lambda}_P^{-2} \mathbf{U}_P^T$$

$$\begin{aligned}
 &= \mathbf{V}_P \boldsymbol{\Lambda}_P \mathbf{U}_P^T \mathbf{U}_P \boldsymbol{\Lambda}_P^{-2} \mathbf{U}_P^T \\
 &= \mathbf{V}_P \boldsymbol{\Lambda}_P \boldsymbol{\Lambda}_P^{-2} \mathbf{U}_P^T \\
 &= \mathbf{V}_P \boldsymbol{\Lambda}_P^{-1} \mathbf{U}_P^T \\
 &= \mathbf{G}_g^{-1}
 \end{aligned} \tag{7.46}$$

as required.

Fitting the Data Exactly When $P = N < M$

As before, let the generalized inverse solution \mathbf{m}_g be given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \tag{7.36}$$

Then the predicted data $\hat{\mathbf{d}}$ is given by

$$\begin{aligned}
 \hat{\mathbf{d}} &= \mathbf{G} \mathbf{m}_g \\
 &= \mathbf{G} [\mathbf{G}_g^{-1} \mathbf{d}] \\
 &= \mathbf{G} \mathbf{G}_g^{-1} \mathbf{d} \\
 &= \mathbf{U}_P \boldsymbol{\Lambda}_P \mathbf{V}_P^T \mathbf{V}_P \boldsymbol{\Lambda}_P^{-1} \mathbf{U}_P^T \mathbf{d} \\
 &= \mathbf{U}_P \mathbf{U}_P^T \mathbf{d} \\
 &= \mathbf{d}
 \end{aligned} \tag{7.47}$$

since $\mathbf{U}_P \mathbf{U}_P^T = \mathbf{I}_N$ whenever $P = N$.

Thus, one can fit the data exactly whenever $P = N$. The reason is that \mathbf{U}_0 is empty when $P = N$. That is, \mathbf{U}_P is equal to \mathbf{U} space.

The Generalized Inverse Solution \mathbf{m}_g Lies in \mathbf{V}_P Space

The generalized inverse solution \mathbf{m}_g is given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \tag{7.36}$$

and is a vector in model space. It lies completely in \mathbf{V}_P space. The way to see this is to take the dot product of \mathbf{m}_g with the eigenvectors in \mathbf{V}_0 . If \mathbf{m}_g has no projection in \mathbf{V}_0 space, then it lies completely in \mathbf{V}_P space.

Thus,

$$\begin{aligned}\mathbf{V}_0^T \mathbf{m}_g &= \mathbf{V}_0^T \mathbf{G}_g^{-1} \mathbf{d} \\ &= \mathbf{V}_0^T \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T \mathbf{d} \\ &= \mathbf{0} \\ &\quad (M-P) \times 1\end{aligned}\tag{7.48}$$

since $\mathbf{V}_0^T \mathbf{V}_P = \mathbf{0}$.

Nonuniqueness of the Solution When $P = N < M$

The solution to $\mathbf{Gm} = \mathbf{d}$ is nonunique because \mathbf{V}_0 exists when $P < M$. Let the general solution \mathbf{m} to $\mathbf{Gm} = \mathbf{d}$ be given by

$$\hat{\mathbf{m}} = \mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{v}_i\tag{7.49}$$

That is, the general solution is given by the generalized inverse solution \mathbf{m}_g plus a linear combination of the eigenvectors in \mathbf{V}_0 space, where α_i are constants. The predicted data for the general case is given by

$$\begin{aligned}\mathbf{G}\hat{\mathbf{m}} &= \mathbf{G} \left[\mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{v}_i \right] \\ &= \mathbf{G}\mathbf{m}_g + \sum_{i=P+1}^M \alpha_i \mathbf{G}\mathbf{v}_i\end{aligned}\tag{7.50}$$

When \mathbf{G} operates on a vector in \mathbf{V}_0 space, however, it returns a zero vector. That is,

$$\begin{aligned}\mathbf{G}\mathbf{V}_0 &= \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T \mathbf{V}_0 \\ &= \mathbf{0} \\ &\quad N \times (M-P)\end{aligned}\tag{7.51}$$

which follows from the fact that the eigenvectors in \mathbf{V}_P are perpendicular to the eigenvectors in \mathbf{V}_0 . Thus,

$$\begin{aligned}\mathbf{G}\hat{\mathbf{m}} &= \mathbf{G}\mathbf{m}_g + \mathbf{0} \\ &= \mathbf{d}\end{aligned}\tag{7.52}$$

Now, consider the length squared of $\hat{\mathbf{m}}$

$$\|\hat{\mathbf{m}}\|^2 = \|\mathbf{m}_g\|^2 + \sum_{i=P+1}^M \alpha_i^2 \quad (7.53)$$

which follows from the fact that $[\mathbf{v}_i]^T \mathbf{v}_j = \delta_{ij}$.

$$\|\hat{\mathbf{m}}\|^2 \geq \|\mathbf{m}_g\|^2 \quad (7.54)$$

That is, \mathbf{m}_g , the generalized inverse solution, is the smallest of all possible solutions to $\mathbf{G}\mathbf{m} = \mathbf{d}$. This is precisely what was stated at the beginning of this section: the generalized inverse solution is equivalent to the minimum length solution when $P = N < M$.

It is Possible to Write \mathbf{G}_g^{-1} Without \mathbf{V}_P When $P = N < M$

To see this, we write the generalized inverse operator as the minimum length operator and use singular-value decomposition. That is,

$$\begin{aligned} \mathbf{G}_g^{-1} &= \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} \\ &\vdots \\ &= \mathbf{G}^T \mathbf{U}_P \mathbf{\Lambda}_P^{-2} \mathbf{U}_P^T \end{aligned} \quad (7.55)$$

Typically, this shortcut is not used because knowledge of \mathbf{V}_P space is useful in the interpretation of the results.

7.2.5 Class IV: $P < \min(N, M)$

This is the class of problems for which neither least squares nor minimum length operators exist. That is, the least squares operator

$$\mathbf{G}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \quad (3.28)$$

does not exist because $[\mathbf{G}^T \mathbf{G}]^{-1}$ exists only when $P = M$. Similarly, the minimum length operator

$$\mathbf{G}_{ML} = \mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.69)$$

does not exist because $[\mathbf{G}\mathbf{G}^T]^{-1}$ exists only when $P = N$.

For $P < \min(N, M)$ we have

1. \mathbf{V}_0 is not empty.
2. \mathbf{U}_0 is not empty.

In this environment the solution is both nonunique (because \mathbf{V}_0 exists), and it is impossible to fit the data exactly unless the compatibility equations (Equations 6.69) are satisfied. That is, it is impossible to fit the data exactly unless the data have no projection onto \mathbf{U}_0 space.

The generalized inverse operator cannot be further reduced and is given by Equation (7.9)

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \quad (7.9)$$

The generalized inverse operator \mathbf{G}_g^{-1} simultaneously minimizes the misfit vector $\mathbf{d} - \hat{\mathbf{d}}$ in data space *and* the solution length $\|\mathbf{m}_g\|^2$ in model space.

In summary, in this section we have shown that the generalized inverse operator \mathbf{G}_g^{-1} reduces to

1. The exact inverse when $P = N = M$.
2. The least squares inverse $[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$ when $P = M < N$.
3. The minimum length inverse $\mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1}$ when $P = N < M$.

Since we have shown that the generalized inverse is equivalent to the exact, least squares, and minimum length operators when they exist, it is worth comparing the way the solution \mathbf{m}_g is written. In the least squares or unique inverse environment for example, we would then write

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} \quad (7.56)$$

but in the minimum length environment we would write

$$\mathbf{m}_g = \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \quad (7.57)$$

which explicitly includes a dependence on the prior estimate $\langle \mathbf{m} \rangle$. It is somewhat disconcerting to have to carry around two forms of the solution for the generalized inverse. Consider what happens, however if we use Equation (7.57) for the unique or least squares environment. Then

$$\begin{aligned} \mathbf{m}_g &= \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \\ &= \langle \mathbf{m} \rangle + \mathbf{G}_g^{-1} \mathbf{d} - \mathbf{G}_g^{-1} \mathbf{G} \langle \mathbf{m} \rangle \\ &= \mathbf{G}_g^{-1} \mathbf{d} + [\mathbf{I}_M - \mathbf{G}_g^{-1} \mathbf{G}] \langle \mathbf{m} \rangle \end{aligned} \quad (7.58)$$

For the unique inverse environment, $\mathbf{G}_g^{-1} \mathbf{G} = \mathbf{I}_M$, and hence Equation (7.58) reduces to Equation (7.56). For the least squares environment, we have

$$\mathbf{G}_g^{-1} \mathbf{G} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{G} = \mathbf{I}_M \quad (7.59)$$

and hence Equation (7.58) again reduces to Equation (7.56). The unique inverse and least squares environments thus have no dependence on $\langle \mathbf{m} \rangle$. Equation (7.57), however, is true for the generalized inverse in all environments and is thus adopted as the general form of the generalized inverse solution \mathbf{m}_g .

In the next section we will introduce measures of the quality of the generalized inverse operator. These will include the *model resolution matrix* \mathbf{R} , the *data resolution matrix* \mathbf{N} (also called the *data information density matrix*), and the *unit (model) covariance matrix* $[\text{cov}_u \mathbf{m}]$.

7.3 Measures of Quality for the Generalized Inverse

7.3.1 Introduction

In this section three measures of quality for the generalized inverse will be considered. They are

1. The $M \times M$ model resolution matrix \mathbf{R}
2. The $N \times N$ data resolution matrix \mathbf{N}
3. The $M \times M$ unit covariance matrix $[\text{cov}_u \mathbf{m}]$

The model resolution matrix \mathbf{R} measures the ability of the inverse operator to uniquely determine the estimated model parameters. The data resolution matrix \mathbf{N} measures the ability of the inverse operator to uniquely determine the data. This is equivalent to describing the importance, or independent information, provided by the data. The two resolution matrices depend upon the partitioning of model and data spaces into $\mathbf{V}_P, \mathbf{V}_0$, and $\mathbf{U}_P, \mathbf{U}_0$ spaces, respectively. Finally, the unit covariance matrix $[\text{cov}_u \mathbf{m}]$ is a measure of how uncorrelated noise with unit variance in the data is mapped into uncertainties in the estimated model parameters.

7.3.2 The Model Resolution Matrix \mathbf{R}

Imagine for the moment that there is some “true” solution \mathbf{m}^{true} that exactly satisfies

$$\mathbf{G}\mathbf{m}^{\text{true}} = \mathbf{d} \quad (7.60)$$

In any inversion, we estimate this true solution with \mathbf{m}^{est} :

$$\mathbf{m}^{\text{est}} = \mathbf{G}_{\text{est}}^{-1} \mathbf{d} \quad (7.61)$$

where $\mathbf{G}_{\text{est}}^{-1}$ is some inverse operator. It is then possible to ask how \mathbf{m}^{est} compares to \mathbf{m}^{true} .

Specifically considering the generalized inverse, we start with Equation (7.61) and replace \mathbf{d} with $\mathbf{G}\mathbf{m}^{\text{true}}$, obtaining

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{G}\mathbf{m}^{\text{true}} \quad (7.62)$$

The model resolution matrix \mathbf{R} is then defined as

$$\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} \quad (7.63)$$

where \mathbf{R} is an $M \times M$ symmetric matrix.

Some Properties of \mathbf{R}

1. $\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T$

Using singular-value decomposition on Equation (7.63), \mathbf{R} can be written as

$$\begin{aligned} \mathbf{R} &= \{ \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \} \{ \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \} \\ &= \mathbf{V}_P \Lambda_P^{-1} \Lambda_P \mathbf{V}_P^T \\ &= \mathbf{V}_P \mathbf{V}_P^T \end{aligned} \quad (7.66)$$

In general, $\mathbf{V}_P \mathbf{V}_P^T \neq \mathbf{I}$. However, if $P = M$, then $\mathbf{V}_P = \mathbf{V}$, and \mathbf{V}_0 is empty. In this case,

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \mathbf{V} \mathbf{V}^T = \mathbf{I}_M$$

since \mathbf{V} is an orthogonal matrix. Thus, the condition for perfect model resolution is that \mathbf{V}_0 be empty, or equivalently that $P = M$.

2. Trace (\mathbf{R}) = $\sum_{i=1}^M r_{ii} = P$, the number of nonzero singular values

Proof: If $\mathbf{R} = \mathbf{I}_M$, then $P = M$ and trace (\mathbf{R}) = M .

For the general case, it is possible to write \mathbf{R} as the product of the following three partitioned matrices:

$$\begin{aligned} \mathbf{R} &= [\mathbf{V}_P | \mathbf{V}_0] \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_P^T \\ \mathbf{V}_0^T \end{bmatrix} \\ &= \mathbf{V} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^T \\ &= \mathbf{V} \mathbf{A} \mathbf{V}^T \end{aligned} \quad (7.67)$$

where no part of \mathbf{V}_0 actually contributes to \mathbf{R} because of the extra zeros in \mathbf{A} .

The trace of \mathbf{A} is equal to P . Note, however, that matrix \mathbf{A} has been obtained from \mathbf{R} by an orthogonal transformation because \mathbf{V} is an orthogonal matrix. Thus, by Equation (2.30d), which states that the trace of a matrix is unchanged by an orthogonal transformation, we conclude that trace $\mathbf{R} = P$, as required. Q.E.D.

Trace (\mathbf{R}) = P implies that \mathbf{G} has enough information to uniquely resolve P aspects of the solution. These aspects are, in fact, the P directions in model space given by the eigenvectors \mathbf{v}_i in \mathbf{V}_P . Whenever a row of \mathbf{R} is equal to a row of the identity matrix \mathbf{I} , then

no part of the associated model parameter \mathbf{m}_i falls in \mathbf{V}_0 space (i.e., it all falls in \mathbf{V}_p space) and that model parameter is perfectly resolved. When \mathbf{R} is not equal to the identity matrix \mathbf{I} , some part of the problem is not perfectly resolved. Sometimes this is acceptable and other times it is not, depending on the problem. Forming new model parameters as linear combinations of the old model parameters is one way to reduce the nonuniqueness of the problem. One way to do this is to form new model parameters by using the eigenvectors \mathbf{v}_i to define the linear combinations. Suppose that $\mathbf{v}_i, i > p$, is given by

$$\mathbf{v}_i = (1 / \sqrt{M})[1, 1, 1, \dots, 1]^T \quad (7.68)$$

This tells us that the average of all the model parameters is resolved, even if the individual model parameters may not be. If we defined a new model parameter as the average of all the old model parameters, it would be perfectly resolved.

If, as is often the case, \mathbf{G} represents some kind of an averaging function, you can attempt to reduce the nonuniqueness of the problem by forming new model parameters that are the sum or average of a subset of the old ones, even without using the full information in \mathbf{V}_p . If the model parameters are discretized versions of a continuous function, such as velocity or density versus depth, you may be able to improve the resolution by combining layers. A rule of thumb in this case is to sum the entries along the diagonal of the resolution matrix \mathbf{R} until you get close to one. At this point, your system is able to resolve one aspect of the solution uniquely. You can try forming a new model parameter as the average of the layer velocities or densities up to this point. Depending on the details of \mathbf{G} , you may have perfect resolution of this average of the old model parameters. Depending on the problem, it may be more useful to uniquely know the average of the model parameters over some depth range than it is to have nonunique estimates of the values over the same range.

3.
$$\sum_{j=1}^M r_{ij}^2 = \sum_{j=1}^M r_{ji}^2 = r_{ii} = \text{“importance” of } i\text{th model parameter}$$

If $r_{ii} = 1$, then the i th model parameter is uniquely resolved, and it is thus said to be very important. If, on the other hand, r_{ii} is very small, then the parameter is poorly resolved and is said to be not very important.

If we further note that \mathbf{R} can be written as

$$\mathbf{R} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_M \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

where \mathbf{r}_i is the i th column of \mathbf{R} , then the estimated solution \mathbf{m}_g from (7.62) can be written as

$$\begin{aligned} \mathbf{m}_g &= \mathbf{R}\mathbf{m}^{\text{true}} \\ &= \mathbf{r}_1 m_1 + \mathbf{r}_2 m_2 + \dots + \mathbf{r}_M m_M \end{aligned}$$

where m_i is the i th component of \mathbf{m}^{true} , which follows from (2.15)–(2.21). That is, the estimated solution vector can also be thought of as the weighted sum of the columns of \mathbf{R} , with the weighting factors being given by the true solution.

The length squared of each column of \mathbf{R} can then be thought of as the importance of m_i in the solution. The length squared of \mathbf{r}_i is given by

$$\|\mathbf{r}_i\|^2 = \sum_{j=1}^M r_{ji}^2 = r_{ii}$$

Thus, the diagonal entries in \mathbf{R} give the importance of each model parameter for the problem.

We will return to the model resolution matrix \mathbf{R} later to show how the generalized inverse is the inverse operator that minimizes the difference between $\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G}$ and \mathbf{I}_M in the least squares sense, and when we discuss the trade-off between resolution and variance.

7.3.3 The Data Resolution Matrix \mathbf{N}

Consider the development of the data resolution matrix \mathbf{N} , which follows closely that of the model resolution matrix \mathbf{R} . The estimated solution, for any inverse operator $\mathbf{G}_{\text{est}}^{-1}$, is given by

$$\mathbf{m}^{\text{est}} = \mathbf{G}_{\text{est}}^{-1} \mathbf{d} \quad (7.69)$$

The predicted data, $\hat{\mathbf{d}}$, for this estimated solution are given by

$$\hat{\mathbf{d}} = \mathbf{G} \mathbf{m}^{\text{est}} \quad (7.70)$$

Replacing \mathbf{m}^{est} in (7.70) with (7.69) gives

$$\hat{\mathbf{d}} = \mathbf{G} \mathbf{G}_{\text{est}}^{-1} \mathbf{d} = \mathbf{N} \mathbf{d} \quad (7.71)$$

where \mathbf{N} is an $N \times N$ matrix called the *data resolution matrix*.

A Specific Example

As a specific example, consider the generalized inverse operator \mathbf{G}_g^{-1} . Then \mathbf{N} is given by

$$\mathbf{N} = \mathbf{G} \mathbf{G}_g^{-1} \quad (7.72)$$

If $\mathbf{N} = \mathbf{I}_N$, then the predicted data $\hat{\mathbf{d}}$ equal the observed data \mathbf{d} , and the observed data can be fit exactly. If $\mathbf{N} \neq \mathbf{I}_N$, then the predicted data are some weighted average of the observed data \mathbf{d} .

Consider the k th element of the predicted data \hat{d}_k

$$\hat{d}_k = \left[\begin{array}{c} \text{kth row of } \mathbf{N} \end{array} \right] \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \quad (7.73)$$

The rows of \mathbf{N} are “windows” through which the observed data are viewed. If the k th row of \mathbf{N} has a 1 in the k th column and zeroes elsewhere, the k th observation is perfectly resolved. We also say that the k th observation, in this case, provides completely independent information. For this reason, \mathbf{N} is sometimes also referred to as the *data information density matrix*. Equation (7.73) shows that the k th predicted datum is a weighted average of all of the observations, with the weighting given by the entries in the k th row of \mathbf{N} . If the k th row of \mathbf{N} has many nonzero elements, then the k th predicted observation depends on the true value of many of the observations, and not just on the k th. The data resolution of the k th observation, then, is said to be poor.

Some Properties of \mathbf{N} for the Generalized Inverse

$$1. \quad \mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} = \mathbf{U}_P\mathbf{U}_P^T$$

Using singular-value decomposition, we have that

$$\begin{aligned} \mathbf{N} = \mathbf{G}\mathbf{G}_g^{-1} &= \mathbf{U}_P\Lambda_P\mathbf{V}_P^T\mathbf{V}_P\Lambda_P^{-1}\mathbf{U}_P^T \\ &= \mathbf{U}_P\mathbf{U}_P^T \end{aligned} \quad (7.74)$$

since $\mathbf{V}\mathbf{V}_P = \mathbf{I}$ (\mathbf{V}_P is semiorthogonal) and $\Lambda_P\Lambda_P^{-1} = \mathbf{I}$.

In general, $\mathbf{U}_P\mathbf{U}_P^T \neq \mathbf{I}_N$. However, if $P = N$, then $\mathbf{U}_P = \mathbf{U}$, and \mathbf{U}_0 is empty. Then, $\mathbf{N} = \mathbf{U}_P\mathbf{U}_P^T = \mathbf{U}\mathbf{U}^T = \mathbf{I}_N$, since \mathbf{U} is itself an orthogonal matrix. Thus, the condition for perfect data resolution is that \mathbf{U}_0 be empty, or that $P = N$.

$$2. \quad \text{trace}(\mathbf{N}) = P$$

The proof follows that of $\text{trace}(\mathbf{R}) = P$ in (7.63):

$$\mathbf{N} = \mathbf{U}_P\mathbf{U}_P^T = [\mathbf{U}_P | \mathbf{U}_0] \begin{bmatrix} \mathbf{I}_P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}_P^T \\ \mathbf{U}_0^T \end{bmatrix}$$

and

$$\text{trace} \begin{bmatrix} \mathbf{I}_P & 0 \\ 0 & 0 \end{bmatrix} = P \quad \text{Q.E.D.}$$

If, for example,

$$n_{11} + n_{22} \approx 1 \quad (7.75a)$$

one might choose to form a new observation d'_1 as a linear combination of d_1 and d_2 , given in the simplest case by

$$d'_1 = d_1 + d_2 \quad (7.75b)$$

The actual linear combination of the two observations that is resolved depends on the eigenvectors in \mathbf{U}_p , or equivalently upon the structure of the data resolution matrix \mathbf{N} . In any case, the new observation d' could provide essentially independent information and could be a way to reduce the computational effort of the inverse problem by reducing \mathbf{N} . In many cases, however, the benefit of being able to average out data errors over the observations is more important than any computational savings that might come from combining observations.

3.
$$\sum_{j=1}^N n_{ij}^2 = \sum_{j=1}^N n_{ji}^2 = n_{ii} = \text{“importance” of the } i\text{th observation} \quad (7.76)$$

That is, the sum of squares of the entries in a row (or column, since \mathbf{N} is symmetric) of \mathbf{N} is equal to the diagonal entry in that row. Thus, as the diagonal entry gets large (close to one), the other entries in that row must become small. As the importance of a particular datum becomes large, the dependence of the predicted datum on other observations must become small.

If we further note that we can write \mathbf{N} as

$$\mathbf{N} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{n}_1 & \mathbf{n}_2 & \cdots & \mathbf{n}_N \\ \vdots & \vdots & & \vdots \end{bmatrix} \quad (7.77)$$

where \mathbf{n}_i is the i th column of \mathbf{N} , then the predicted data $\hat{\mathbf{d}}$ from (7.71) can be written as

$$\begin{aligned} \hat{\mathbf{d}} &= \mathbf{N}\mathbf{d} \\ &= \mathbf{n}_1 d_1 + \mathbf{n}_2 d_2 + \cdots + \mathbf{n}_N d_N \end{aligned} \quad (7.78)$$

where d_i is the i th component of \mathbf{d} . Equation (7.78) follows from (2.15)–(2.21). That is, the predicted data vector can also be thought of as the weighted sum of the columns of \mathbf{N} , with the weighting factors being given by the actual observations.

The length squared of each column of \mathbf{N} can then be thought of as the importance of d_i in the solution. The length squared of \mathbf{n}_i is given by

$$\|\mathbf{n}_i\|^2 = \sum_{j=1}^N n_{ji}^2 = n_{ii} \quad (7.79)$$

Thus, the diagonal entries in \mathbf{N} give the importance of each observation in the solution.

It can also be shown that the generalized inverse operator $\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T$ minimizes the difference between \mathbf{N} and \mathbf{I}_N . Let us now turn our attention to another measure of quality for the generalized inverse, the unit covariance matrix $[\text{cov}_u \mathbf{m}]$.

7.3.4 The Unit (Model) Covariance Matrix $[\text{cov}_u \mathbf{m}]$

Any errors (noise) in the data will be mapped into errors in the estimates of the model parameters. The mapping was first considered in Section 3.7 of Chapter 3. We will now reconsider it from the generalized inverse viewpoint.

Let the error (noise) in the data \mathbf{d} be $\Delta \mathbf{d}$. Then the error in the model parameters due to $\Delta \mathbf{d}$ is given by

$$\Delta \mathbf{m} = \mathbf{G}_g^{-1} \Delta \mathbf{d} \quad (7.80)$$

Step 1. Recall from Equation (2.42c) that the $N \times N$ data covariance matrix $[\text{cov } \mathbf{d}]$ is given by

$$[\text{cov } \mathbf{d}] = \frac{1}{k-1} \sum_{i=1}^k \Delta \mathbf{d}^i [\Delta \mathbf{d}^i]^T \quad (7.81)$$

where k is the number of experiments, and i is the experiment number. The diagonal terms are the data variances and the off-diagonal terms are the covariances.

The data covariance is also written as $\langle \Delta \mathbf{d} \Delta \mathbf{d}^T \rangle$, where $\langle \rangle$ denotes averaging.

Step 2. We seek, then, a model covariance matrix $\langle \Delta \mathbf{m} \Delta \mathbf{m}^T \rangle = [\text{cov } \mathbf{m}]$.

$$\begin{aligned} \langle \Delta \mathbf{m} \Delta \mathbf{m}^T \rangle &= \langle \mathbf{G}_g^{-1} \Delta \mathbf{d} [\mathbf{G}_g^{-1} \Delta \mathbf{d}]^T \rangle \\ &= \langle \mathbf{G}_g^{-1} \Delta \mathbf{d} \Delta \mathbf{d}^T [\mathbf{G}_g^{-1}]^T \rangle \end{aligned} \quad (7.82)$$

\mathbf{G}_g^{-1} is not changing with each experiment, so we can take it outside the averaging, implying

$$\langle \Delta \mathbf{m} \Delta \mathbf{m} \rangle^T = \mathbf{G}_g^{-1} \langle \Delta \mathbf{d} \Delta \mathbf{d}^T \rangle [\mathbf{G}_g^{-1}]^T$$

or

$$[\text{cov } \mathbf{m}] = \mathbf{G}_g^{-1} [\text{cov } \mathbf{d}] [\mathbf{G}_g^{-1}]^T \quad (7.83)$$

The above derivation provides some of the logic behind Equation (2.44c), which was introduced in Chapter 2 as magic.

Step 3. Finally, define a unit (model) covariance matrix $[\text{cov}_u \mathbf{m}]$ by assuming that $[\text{cov} \mathbf{d}] = \mathbf{I}_N$, that is, by assuming that all the data variances are equal to 1 and the covariances are all 0 (uncorrelated data errors). Then

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\text{cov} \mathbf{d}] [\mathbf{G}_g^{-1}]^T \\ &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \end{aligned} \quad (7.84)$$

Some Properties of $[\text{cov}_u \mathbf{m}]$

1. Using singular-value decomposition, we can write the unit model covariance matrix as

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\ &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T [\mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T]^T \\ &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \mathbf{U}_P \Lambda_P^{-1} \mathbf{V}_P^T \\ &= \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T \end{aligned} \quad (7.85)$$

This emphasizes the importance of the size of the singular values λ_i in determining the model parameter variances. As λ_i gets small, the entries in $[\text{cov}_u \mathbf{m}]$ tend to get big (implying large model parameter estimate variances) due to the terms in $1/\lambda^2$

Consider the k th diagonal entry in $[\text{cov}_u \mathbf{m}]$, $[\text{cov}_u \mathbf{m}]_{kk}$, where

$$[\text{cov}_u \mathbf{m}] = \begin{bmatrix} \vdots & \vdots & & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_P \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} 1/\lambda_1^2 & 0 & \cdots & 0 \\ 0 & 1/\lambda_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\lambda_P^2 \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1^T & \cdots \\ \cdots & \mathbf{v}_2^T & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \mathbf{v}_P^T & \cdots \end{bmatrix} \quad (7.86)$$

If we multiply out the first two matrices, we can then identify the kk entry in $[\text{cov}_u \mathbf{m}]$ as the product of the k th row times the k th column of the resulting matrices

$$\begin{aligned} [\text{cov}_u \mathbf{m}]_{kk} &= \begin{bmatrix} v_{11}/\lambda_1^2 & v_{12}/\lambda_2^2 & \cdots & v_{1P}/\lambda_P^2 \\ \vdots & \vdots & & \vdots \\ v_{k1}/\lambda_1^2 & v_{k2}/\lambda_2^2 & \cdots & v_{kP}/\lambda_P^2 \\ \vdots & \vdots & & \vdots \\ v_{M1}/\lambda_1^2 & v_{M2}/\lambda_2^2 & \cdots & v_{MP}/\lambda_P^2 \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{k1} & \cdots & v_{M1} \\ v_{12} & \cdots & v_{k2} & \cdots & v_{M2} \\ \vdots & & \vdots & & \vdots \\ v_{1P} & \cdots & v_{kP} & \cdots & v_{MP} \end{bmatrix} \\ &\quad \uparrow \\ &\quad \text{kth column} \end{aligned}$$

$$= \sum_{i=1}^P \frac{v_{ki}^2}{\lambda_i^2} \quad (7.87)$$

Thus, as λ_i gets small, $[\text{cov}_{\mathbf{u}} \mathbf{m}]_{kk}$ will get large if v_{ki} is not zero. Recall that v_{ki} is the k th component in the i th eigenvector \mathbf{v}_i in \mathbf{V}_P . Thus, it is the combination of λ_i getting small, and \mathbf{v}_i having a nonzero component in the k th row, that makes the variance for the k th model parameter potentially very large.

2. Even if the data covariance is diagonal (i.e., all the observations have errors that are uncorrelated), $[\text{cov}_{\mathbf{u}} \mathbf{m}]$ *need not be* diagonal. That is, the model parameter estimates may well have nonzero covariances, even though the data have zero covariances.

For example, from Equation (7.87) above, we can see that

$$[\text{cov}_{\mathbf{u}} \mathbf{m}]_{1k} = \begin{bmatrix} v_{11}/\lambda_1^2 & v_{12}/\lambda_2^2 & \cdots & v_{1P}/\lambda_P^2 \end{bmatrix} \begin{bmatrix} v_{k1} \\ v_{k2} \\ \vdots \\ v_{kP} \end{bmatrix} = \sum_{i=1}^P \frac{v_{1i}v_{ki}}{\lambda_i^2} \quad (7.88)$$

Note that the term in the above equation

$$\sum_{i=1}^P \frac{v_{1i}v_{ki}}{\lambda_i^2}$$

is *not* the dot product of two *columns* of \mathbf{V}_P . In fact, even if the numerator were the dot product between columns (i.e., $v_{1i}v_{ik}$), the fact that every term is divided by λ_i^2 would likely yield something other than 1. The numerator is the dot product of two *rows* of \mathbf{V}_P and is likely nonzero anyway.

3. Notice that $[\text{cov}_{\mathbf{u}} \mathbf{m}]$ is a function of the forward problem as expressed in \mathbf{G} , and *not* a function of the actual data. Thus, it can be useful for experimental design.

7.3.5 Combining \mathbf{R} , \mathbf{N} , $[\text{cov}_{\mathbf{u}} \mathbf{m}]$

Note that, in general, \mathbf{G} , \mathbf{G}_g^{-1} , \mathbf{R} , \mathbf{N} , and $[\text{cov}_{\mathbf{u}} \mathbf{m}]$ can be written in terms of singular-value decomposition as

1. $\mathbf{G} = \mathbf{U}_P \mathbf{\Lambda}_P \mathbf{V}_P^T$
2. $\mathbf{G}_g^{-1} = \mathbf{V}_P \mathbf{\Lambda}_P^{-1} \mathbf{U}_P^T$
3. $\mathbf{R} = \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{V}_P \mathbf{V}_P^T$
4. $\mathbf{N} = \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{U}_P \mathbf{U}_P^T$

$$\begin{aligned}
 5. \quad [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\
 &= \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T
 \end{aligned}$$

Case I: $P = M = N$

$$\begin{aligned}
 \mathbf{R} &= \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{I}_M, \text{ since } \mathbf{G}_g^{-1} = \mathbf{G}^{-1} \\
 \mathbf{N} &= \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{G} \mathbf{G}^{-1} = \mathbf{I}_N, \text{ since } \mathbf{G}_g^{-1} = \mathbf{G}^{-1} \\
 [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\
 &= \mathbf{V} \Lambda^{-1} \mathbf{U}^T [\mathbf{V} \Lambda^{-1} \mathbf{U}^T]^T \\
 &= \mathbf{V} \Lambda^{-1} \mathbf{U}^T \mathbf{U} \Lambda^{-1} \mathbf{V}^T = \mathbf{V} \Lambda^{-2} \mathbf{V}^T
 \end{aligned}$$

Case II: $P = M < N$ (Least Squares)

$$\begin{aligned}
 \mathbf{G}_g^{-1} &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \\
 \mathbf{R} &= \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{V}_P \mathbf{V}_P^T = \mathbf{V} \mathbf{V}^T = \mathbf{I}_M \text{ since } P = M \\
 \mathbf{N} &= \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{G} \{[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T\} = (\text{using SVD } \dots) = \mathbf{U}_P \mathbf{U}_P^T \\
 [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\
 &= [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \{[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T\}^T \\
 &= (\text{using SVD } \dots) = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T
 \end{aligned}$$

Case III: $P = N < M$ (Minimum Length)

$$\begin{aligned}
 \mathbf{G}_g^{-1} &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T = \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1} \\
 \mathbf{R} &= \mathbf{G}_g^{-1} \mathbf{G} = \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1} \mathbf{G} = (\text{using SVD } \dots) = \mathbf{V}_P \mathbf{V}_P^T \\
 \mathbf{N} &= \mathbf{G} \mathbf{G}_g^{-1} = \mathbf{G} \{ \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1} \} = (\text{using SVD } \dots) = \mathbf{U}_P \mathbf{U}_P^T = \mathbf{U} \mathbf{U}^T = \mathbf{I}_N \\
 [\text{cov}_u \mathbf{m}] &= \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T \\
 &= \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1} \{ \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1} \}^T \\
 &= (\text{using SVD } \dots) = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T
 \end{aligned}$$

Case IV: $P < \min(M, N)$ (General Case)

This is just the general case.

7.3.6 An Illustrative Example

Consider a system of equations $\mathbf{G}\mathbf{m} = \mathbf{d}$ given by

$$\begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.01 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 4.10 \end{bmatrix} \quad (7.89)$$

Doing singular-value decomposition, one finds

$$\lambda_1 = 3.169$$

$$\lambda_2 = 0.00316$$

$$\mathbf{U}_P = \mathbf{U} = \begin{bmatrix} 0.446 & -0.895 \\ 0.895 & 0.446 \end{bmatrix}$$

$$\mathbf{V}_P = \mathbf{V} = \begin{bmatrix} 0.706 & -0.709 \\ 0.709 & 0.706 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} = \mathbf{I}_2$$

$$\mathbf{N} = \mathbf{U}_P \mathbf{U}_P^T = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} = \mathbf{I}_2$$

$$\mathbf{G}_g^{-1} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix}$$

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix} \begin{bmatrix} 2.0 \\ 4.1 \end{bmatrix} = \begin{bmatrix} -8.0 \\ 10.0 \end{bmatrix} \quad (7.90)$$

Note that the solution has perfect model resolution ($\mathbf{R} = \mathbf{I}$, and hence the solution is unique) and perfect data resolution ($\mathbf{N} = \mathbf{I}$, and hence the data can be fit exactly). Note also that $P = N = M$, and the generalized inverse is, in fact, the unique mathematical inverse.

This solution is, however, essentially meaningless if the data contain even a small amount of noise. To see this, consider the unit covariance matrix $[\text{cov}_u \mathbf{m}]$ for this case:

$$[\text{cov}_u \mathbf{m}] = \mathbf{G}_g^{-1} [\mathbf{G}_g^{-1}]^T = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P^T = \mathbf{V}_P \begin{bmatrix} 0.0996 & 0 \\ 0 & 100,300.9 \end{bmatrix} \mathbf{V}_P^T$$

$$= \begin{bmatrix} 50,401 & -50,200 \\ -50,200 & 50,000 \end{bmatrix} \quad (7.91)$$

These are very large covariances for m_1 and m_2 , which indicate that the solution, while unique and fitting the data perfectly, is very unstable, or sensitive to noise in the data. For example, suppose that d_2 is 4.0 instead of 4.1 (2.5% error). Then the generalized inverse solution \mathbf{m}_g is given by

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 201 & -100 \\ -200 & 100 \end{bmatrix} \begin{bmatrix} 2.0 \\ 4.0 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \end{bmatrix} \quad (7.92)$$

That is, errors of less than a few percent in \mathbf{d} result in errors on the order of several hundred percent in \mathbf{m}_g . Whenever small changes in \mathbf{d} result in large changes in \mathbf{m}_g , the problem is considered unstable. In this particular problem, if a solution is desired with a standard deviation of order 0.1, then the data standard deviations must be less than about 5×10^{-4} !

Another way of quantifying the instability of the inversion is with the *condition number*, defined as

$$\text{condition number} = \lambda_{\max} / \lambda_{\min} \quad (7.93)$$

For this particular problem, the condition number is approximately 1000, which indicates considerable instability. The condition number, by itself, can be misleading. If a problem has two singular values λ_1 and λ_2 , with $\lambda_1 = 1,000,000$ and $\lambda_2 = 1000$, then $\lambda_1/\lambda_2 = 1000$. This problem, however, is very stable with changes of length order one in the data (do you see why?). If, however, $\lambda_1 = 0.001$ and $\lambda_2 = 0.000001$, then $\lambda_1/\lambda_2 = 1000$, and unit length changes in the data will cause large changes in the solution. In addition to just the condition number, the absolute size of the singular values is important, especially compared to the size of the possible noise in the data.

In order to gain a better understanding of the origin of the instability, one must consider the structure of the \mathbf{G} matrix itself. For the present example, inspection shows that the columns, or rows, of \mathbf{G} are very nearly parallel to one another. For example, the angle between the vectors given by the columns is 0.114° , obtained by taking the dot product of the two columns. The two columns of \mathbf{G} span the two dimensional data space, and hence the data resolution is perfect, but the fact that they are nearly parallel leads to a significant instability.

It is not a coincidence, therefore, that the data eigenvector associated with the larger singular value, $\mathbf{u}_1 = [0.446, 0.895]^T$, is essentially parallel to the common direction given by the columns of \mathbf{G} . Nor is it a coincidence that \mathbf{u}_2 , associated with the smaller singular value, is perpendicular to the almost uniform direction given by the columns of \mathbf{G} . The eigenvector \mathbf{u}_1 represents a stable direction in data space as far as noise is concerned, while \mathbf{u}_2 represents an unstable direction in data space as far as noise is concerned. Noise in data space parallel to \mathbf{u}_1 will be damped by $1/\lambda_1$, while noise parallel to \mathbf{u}_2 will be amplified by $1/\lambda_2$.

Similar arguments can be made about the rows of \mathbf{G} , which lie in model space. That is, \mathbf{v}_1 is essentially parallel to the almost uniform direction given by the rows of \mathbf{G} , while \mathbf{v}_2 is essentially perpendicular to the direction given by the rows of \mathbf{G} . Noise parallel to \mathbf{u}_1 , when operated on by the generalized inverse, creates noise in the solution parallel to \mathbf{v}_1 , while noise parallel to \mathbf{u}_2 creates noise parallel to \mathbf{v}_2 . Thus, \mathbf{v}_2 is the unstable direction in model space.

Methods to stabilize the model parameter variances will be considered in a later section, but it will also be shown that any gain in stability is obtained at a cost in resolution. First, however, we will introduce ways to quantify \mathbf{R} , \mathbf{N} , and $[\text{cov}_u \mathbf{m}]$. We will return to the above example and show specifically how stability can be enhanced while resolution is lost.

7.4 Quantifying the Quality of \mathbf{R} , \mathbf{N} , and $[\text{cov}_u \mathbf{m}]$

7.4.1 Introduction

In the proceeding sections we have shown that the model resolution matrix \mathbf{R} , the data resolution matrix \mathbf{N} , and the unit model covariance matrix $[\text{cov}_u \mathbf{m}]$ can be very useful, at least in a qualitative way, in assessing the quality of a particular inversion. In this section, we will quantify these measures of quality, and show that the generalized inverse is the inverse that gives the best possible model and data resolution.

First, consider the following definitions (see Menke, page 68):

$$\text{spread}(\mathbf{R}) = \|\mathbf{R} - \mathbf{I}\|_2^2 = \sum_{i=1}^M \sum_{j=1}^M [r_{ij} - \delta_{ij}]^2 \quad (7.94)$$

$$\text{spread}(\mathbf{N}) = \|\mathbf{N} - \mathbf{I}\|_2^2 = \sum_{i=1}^N \sum_{j=1}^N [n_{ij} - \delta_{ij}]^2 \quad (7.95)$$

and

$$\text{size}([\text{cov}_u \mathbf{m}]) = \sum_{i=1}^M [\text{cov}_u \mathbf{m}]_{ii} \quad (7.96)$$

The spread function measures how different \mathbf{R} (or \mathbf{N}) is from an identity matrix. If \mathbf{R} (or \mathbf{N}) = \mathbf{I} , then $\text{spread}(\mathbf{R})$ (or \mathbf{N}) = 0. The size function is the trace of the unit model covariance matrix, which gives the sum of the model parameter variances.

We can now look at the spread and size functions for various classes of problems.

7.4.2 Classes of Problems

Class I: $P = N = M$

$\text{spread}(\mathbf{R}) = \text{spread}(\mathbf{N}) = 0$
 $\text{size}([\text{cov}_u \mathbf{m}])$

perfect model and data resolution
 depends on the size of the singular values

Class II: $P = M < N$ (Least Squares)

spread (\mathbf{R}) = 0	perfect model resolution
spread (\mathbf{N}) \neq 0	data not all independent
size ($[\text{cov}_u \mathbf{m}]$)	depends on the size of the singular values

Class III: $P = N < M$ (Minimum Length)

spread (\mathbf{R}) \neq 0	nonunique solution
spread (\mathbf{N}) = 0	perfect data resolution
size ($[\text{cov}_u \mathbf{m}]$)	depends on the size of the singular values

Class IV: $P < \min(N, M)$ (General Case)

spread (\mathbf{R}) \neq 0	nonunique solution
spread (\mathbf{N}) \neq 0	data not all independent
size ($[\text{cov}_u \mathbf{m}]$)	depends on the size of the singular values

We also note that the position of an off-diagonal nonzero entry in \mathbf{R} or \mathbf{N} does not affect the spread. This is as it should be if the model parameters and data have no physical ordering.

7.4.3 Effect of the Generalized Inverse Operator \mathbf{G}_g^{-1}

We are now in a position to show that the generalized inverse operator \mathbf{G}_g^{-1} gives the best possible \mathbf{R} , \mathbf{N} matrices in terms of minimizing the spread functions as defined in (7.94)–(7.95). Menke (pp. 68–70) does this for the $P = M < N$ case, and less fully for the $P = N < M$ case. Consider instead, the more general derivation (after Jackson, 1972). For any estimate of the inverse operator $\mathbf{G}_{\text{est}}^{-1}$, the model resolution matrix \mathbf{R} is given by

$$\begin{aligned} \mathbf{R} &= \mathbf{G}_{\text{est}}^{-1} \mathbf{G} \\ &= \mathbf{G}_{\text{est}}^{-1} \mathbf{U}_P \Lambda_P \mathbf{V}_P^T \\ &= \mathbf{B} \mathbf{V}_P^T \end{aligned} \tag{7.97}$$

where $\mathbf{B} = \mathbf{G}_{\text{est}}^{-1} \mathbf{U}_P \Lambda_P$. From (2.15)–(2.21), each row of \mathbf{R} will be a linear combination of the rows of \mathbf{V}_P^T , or equivalently a linear combination of the columns of \mathbf{V}_P . The weighting factors are determined by \mathbf{B} , which depends on the choice of the inverse operator.

The goal, then, is to choose an inverse operator that will make \mathbf{R} most like the identity matrix \mathbf{I} in the sense of minimizing spread (\mathbf{R}). Define \mathbf{b}_k^T as the k th row of \mathbf{B} , and \mathbf{d}_k^T as the k th row of \mathbf{I}_M . We seek \mathbf{b}_k^T as the least squares solution to

$$\begin{matrix} \mathbf{b}_k^T & \mathbf{V}_P^T & = & \mathbf{d}_k^T \\ 1 \times P & P \times M & & 1 \times M \end{matrix} \tag{7.98}$$

Taking the transposes implies

$$\begin{matrix} \mathbf{V}_P & \mathbf{b}_k & = & \mathbf{d}_k \\ M \times P & P \times 1 & & M \times 1 \end{matrix} \quad (7.99)$$

Equation (7.99) can be solved with the least squares operator [see (3.19)] as

$$\begin{aligned} \mathbf{b}_k &= [\mathbf{V}_P^T \mathbf{V}_P]^{-1} \mathbf{V}_P^T \mathbf{d}_k \\ &= \mathbf{I}^{-1} \mathbf{V}_P^T \mathbf{d}_k \\ &= \mathbf{V}_P^T \mathbf{d}_k \end{aligned} \quad (7.100)$$

Taking the transpose of (7.100) gives

$$\mathbf{b}_k^T = \mathbf{d}_k^T \mathbf{V}_P \quad (7.101)$$

Writing this out specifically, we have

$$[b_{k1} \quad b_{k2} \quad \cdots \quad b_{kP}] = [0 \quad \cdots \quad 0 \quad \underset{\substack{\uparrow \\ k}}{1} \quad 0 \quad \cdots \quad 0] \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1P} \\ v_{21} & v_{22} & \cdots & v_{2P} \\ \vdots & \vdots & & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MP} \end{bmatrix} \quad (7.102)$$

Looking at (7.102), we see that the i th entry in \mathbf{b}_k^T is given by

$$b_{ki} = v_{ki} \quad (7.103)$$

That is, each element in the k th row of \mathbf{B} is the corresponding element in the k th row of \mathbf{V}_P . Or, simply put, the k th row of \mathbf{B} is given by the k th row of \mathbf{V}_P .

Making similar arguments for each row of \mathbf{B} gives us

$$\mathbf{B} = \mathbf{V}_P \quad (7.104)$$

Substituting \mathbf{B} back into (7.97) gives

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T \quad (7.105)$$

This is, however, exactly the model resolution matrix for the generalized inverse, given in (7.65). Thus, we have shown that the generalized inverse is the operator with the best model resolution in the sense that the least squares difference between \mathbf{R} and \mathbf{I}_M is minimized. Very similar arguments can be made that show that the generalized inverse is the operator with the best data resolution in the sense that the least squares difference between \mathbf{N} and \mathbf{I}_N is minimized.

In cases where the model parameters or data have a natural ordering, such as a discretization of density versus depth (for model parameters) or gravity measurements along a profile (for data), we might want to modify the definition of the spread functions in (7.94)–(7.95). One such

modification leads to the Backus–Gilbert inverse. A modified spread function is defined by

$$\text{spread}(\mathbf{R}) = \sum_{i=1}^M \sum_{j=1}^M W(i, j) [r_{ij} - \delta_{ij}]^2 \quad (7.106)$$

where $W(i, j) = (i - j)^2$. This gives more weight (penalty) to entries far from the diagonal. It has the effect, however, of canceling out any $i = j$ contribution to the spread. To handle this, a constraint equation is added and satisfied by the use of Lagrange multipliers. The constraint equation is given by

$$\sum_{j=1}^M r_{ij} = 1 \quad (7.107)$$

This ensures that not all entries in the row of \mathbf{R} are allowed to go to zero, which would minimize the spread in (7.106). The inverse operator based on (7.106) is called the Backus–Gilbert inverse, first developed for continuous (rather than discrete) problems.

7.5 Resolution Versus Stability

7.5.1 Introduction

We will see in this section that stability can be improved by removing small singular values from an inversion. We will also see, however, that this reduces the resolution. There is an unavoidable trade-off between solution stability and resolution.

Recall the example from Equation (7.89)

$$\begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.01 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 4.10 \end{bmatrix} \quad (7.89)$$

The singular values, eigenvector matrices, generalized inverse, and other relevant matrices are given in Equation (7.90).

One option is to arbitrarily set $\lambda_2 = 0$. Then P is reduced from 2 to 1, and

$$\mathbf{U}_P = \begin{bmatrix} 0.446 \\ 0.895 \end{bmatrix}$$

$$\mathbf{V}_P = \begin{bmatrix} 0.706 \\ 0.709 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{V}_P \mathbf{V}_P^T = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{N} = \mathbf{U}_P \mathbf{U}_P^T = \begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T = \begin{bmatrix} 0.099 & 0.199 \\ 0.100 & 0.200 \end{bmatrix}$$

$$[\text{cov}_u \mathbf{m}] = \mathbf{V}_P \Lambda_P^{-2} \mathbf{V}_P = \begin{bmatrix} 0.0496 & 0.0498 \\ 0.0498 & 0.0500 \end{bmatrix}$$

$$\mathbf{m}_g = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 1.016 \\ 1.020 \end{bmatrix}$$

and

$$\hat{\mathbf{d}} = \mathbf{G} \mathbf{m}_g = \begin{bmatrix} 2.04 \\ 4.08 \end{bmatrix} \quad (7.108)$$

First, note that the size of the unit model covariance matrix has been significantly reduced, indicating a dramatic improvement in stability in the solution. The model parameter variances are order 0.05 for data with unit variance.

Second, note that the fit to the data, while not perfect, is fairly close. The misfits for d_1 and d_2 are, at most, 2%.

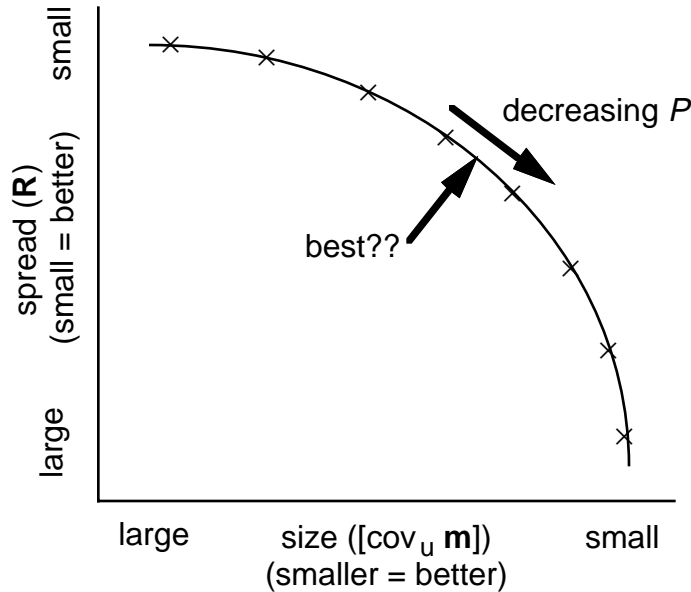
Third, however, note that both model and data resolution have been degraded from perfect resolution when both singular values were retained. In fact, \mathbf{R} now indicates that the estimates for both m_1 and m_2 are given by the average of the true values of m_1 and m_2 . That is, $0.706m_1$ plus $0.704m_2$ is perfectly resolved, but there is no information about the difference. This can also be seen by examining \mathbf{V}_P , which points in the $[0.706, 0.709]^T$ direction in model space. This is the only direction in model space that can be resolved.

Recall on page 165 that when d_2 was changed from 4.1 to 4.0, the solution changed from $[-8, 10]^T$ to $[2, 0]^T$. The sum of m_1 and m_2 remained constant, but the difference changed significantly. If d_2 is changed from 4.1 to 4.0 now, the solution is $[0.998, 1.000]^T$, very close to the solution with $d_2 = 4.1$.

In some cases, knowing the sum of m_1 and m_2 may be useful, such as when \mathbf{m} gives the velocity of some layered structure. Then knowing the average velocity, even if the individual layer velocities cannot be resolved, may be useful. In any case, we have shown that the original decomposition, with two nonzero singular values, was so unstable that the solution, while unique, was essentially meaningless.

The data resolution matrix \mathbf{N} indicates that the second observation is more important than the first (0.8 versus 0.2 along the diagonal). This can be seen either from noting that the second row of either column of \mathbf{G} is larger than the first row, and \mathbf{U}_P is formed as a linear combination of the columns of \mathbf{G} , or by looking at \mathbf{U}_P , which points in the $[0.446, 0.895]^T$ direction in data space.

Another way to look at the trade-off is by plotting resolution versus stability as shown below:



As P is decreased, by setting small singular values to zero, the resolution degrades while the stability increases. Sometimes it is possible to pick an optimal cut-off value for small singular values based on this type of graph.

7.5.2 \mathbf{R} , \mathbf{N} , and $[\text{cov}_u \mathbf{m}]$ for Nonlinear Problems

The resolution matrices and the unit model covariance matrix are also useful in a nonlinear analysis, although the interpretations are somewhat different than they are for the linear case.

Model Resolution Matrix \mathbf{R}

In the linear case the solution is unique whenever $\mathbf{R} = \mathbf{I}$. For the nonlinear problem, a unique solution is not guaranteed, even if $\mathbf{R} = \mathbf{I}$. In fact, no solution may exist, even when $\mathbf{R} = \mathbf{I}$. Consider the following simple nonlinear problem:

$$m^2 = d_1 \tag{7.109}$$

With a single model parameter and a single observation, we have $P = M = N$. Thus, $\mathbf{R} = \mathbf{I}$ at every iteration. If $d_1 = 4$, the process will iterate successfully to the solution $m_1 = 2$ unless, by chance, the iterative process ever gives m_1 exactly equal to zero, in which case the inverse is undefined. However, if d_1 is negative, there is no real solution, and the iterative process will never converge to an answer, even though $\mathbf{R} = \mathbf{I}$.

The uniqueness of nonlinear problems also depends on the existence of local minima. It is always a good idea in nonlinear problems to explore solution space to make sure that the solution obtained corresponds to a global minima. Take, for example, the following case with two observations and two model parameters:

$$\begin{aligned}
 m_1^4 + m_2^2 &= 2 \\
 m_1^2 + m_2^4 &= 2
 \end{aligned}
 \tag{7.110}$$

This simple set of two nonlinear equations in two unknowns has $\mathbf{R} = \mathbf{I}$ almost everywhere in solution space. By inspection, however, there are four solutions that fit the data exactly, given by $[m_1, m_2]^T = [1, 1]^T, [1, -1]^T, [-1, 1]^T,$ and $[-1, -1]^T,$ respectively.

To see the role of the model resolution matrix for a nonlinear analysis, recall Equations (4.12)–(4.16), where, for example,

$$\Delta \mathbf{c} = \mathbf{G} \Delta \mathbf{m}
 \tag{4.13}$$

and where $\Delta \mathbf{c}$ is the misfit to the data, given by the observed minus the predicted data, $\Delta \mathbf{m}$ are changes to the model at this iteration, and \mathbf{G} is the matrix of partial derivatives of the forward equations with respect to the model parameters. If $\mathbf{R} = \mathbf{I}$ at the solution, then the changes $\Delta \mathbf{m}$ are perfectly resolved in the close vicinity of the solution. If $\mathbf{R} \neq \mathbf{I}$, then there will be directions in model space (corresponding to \mathbf{V}_0) that do not change the predicted data, and hence the fit to the data. All of this analysis, of course, is based on the linearization of a nonlinear problem in the vicinity of the solution. The analysis of \mathbf{R} is only as good as the linearization of the nonlinear problem. If the solution is very nonlinear at the solution, the validity of conclusions based on an analysis of \mathbf{R} may be suspect.

Note also that \mathbf{R} , which depends on both \mathbf{G} and the inverse operator, may change during the iterative process. For example, in the Equation (7.110) above, we noted that $\mathbf{R} = \mathbf{I}$ almost everywhere in solution space. At the point given by $[m_1, m_2]^T = [0.7071, 0.7071]^T,$ you may verify for yourselves that all four entries in the \mathbf{G} matrix of partial derivatives are equal to 1. In this case, P is reduced from 2 to 1. The next iteration, however, will take the solution to somewhere else where the resolution matrix is again an identity matrix. The analysis of \mathbf{R} is thus generally reserved for the final iteration at the solution. At intermediate steps, \mathbf{R} determines whether there is a unique *direction* in which to move toward the solution. Since the path to the solution is less critical than the final solution, little emphasis is generally placed on \mathbf{R} during the iterative process.

The generalized inverse operator, which finds the minimum length solution for $\Delta \mathbf{m}$, finds the smallest possible change to the linearized problem to minimize the misfit to the data. This is a benefit because large changes in $\Delta \mathbf{m}$ will take the estimated parameter values farther away from the region where the linearization of the problem is valid.

Data Resolution Matrix \mathbf{N}

In the linear case, $\mathbf{N} = \mathbf{I}$ implies perfectly independent (and resolved) data. In the nonlinear case, $\mathbf{N} = \mathbf{I}$ implies that the misfit $\Delta \mathbf{c}$, and not necessarily the data vector \mathbf{d} itself, is perfectly resolved for the linearized problem. If $\mathbf{N} \neq \mathbf{I}$, then any part of the misfit $\Delta \mathbf{c}$ that lies in \mathbf{U}_0 space will not contribute to changes in the model parameter estimates. In the vicinity of the solution, if $\mathbf{N} = \mathbf{I}$, then data space is completely resolved, and the misfit should typically go to zero. If $\mathbf{N} \neq \mathbf{I}$ at the solution, then there may be a part of the data that cannot be fit. But, even if $\mathbf{N} = \mathbf{I}$, there is no guarantee that there is any solution that will fit the data exactly. Recall the example in Equation (7.109) above where $\mathbf{N} = \mathbf{I}$ everywhere. If d_1 is negative, no real solution can be found that fits the data exactly.

As with the model resolution matrix, \mathbf{N} is most useful at the solution and less useful during the iterative process. Also, it should always be recalled that the analysis of \mathbf{N} is only as valid as the linearization of the problem.

The Unit Model Covariance Matrix [cov_u \mathbf{m}]

For a linear analysis, the unit model covariance provides variance estimates for the model parameters assuming unit, uncorrelated data variances. For the nonlinear case, the unit model covariance matrix provides variance estimates for the model parameter changes $\Delta\mathbf{m}$. At the solution, these can be interpreted as variances for the model parameters themselves, as long as the problem is not too nonlinear. Along the iterative path, and at the final solution, the unit covariance matrix provides an estimate of the stability of the process. If the variances are large, then there must be small singular values, and the misfit may be mapped into large changes in the model parameters. Analysis of particular model parameter variances is usually reserved for the final iteration. As with both the resolution matrices, the model parameter variance estimates are based on the linearized problem, and are only as good as the linearization itself.

Consider a simple $N = M = 1$ nonlinear forward problem given by

$$d = m^{1/3} \tag{7.111}$$

The inverse solution (exact, or equivalently the generalized inverse) is of course given by

$$m = d^3 \tag{7.112}$$

These relationships are shown in the figures on the next page.

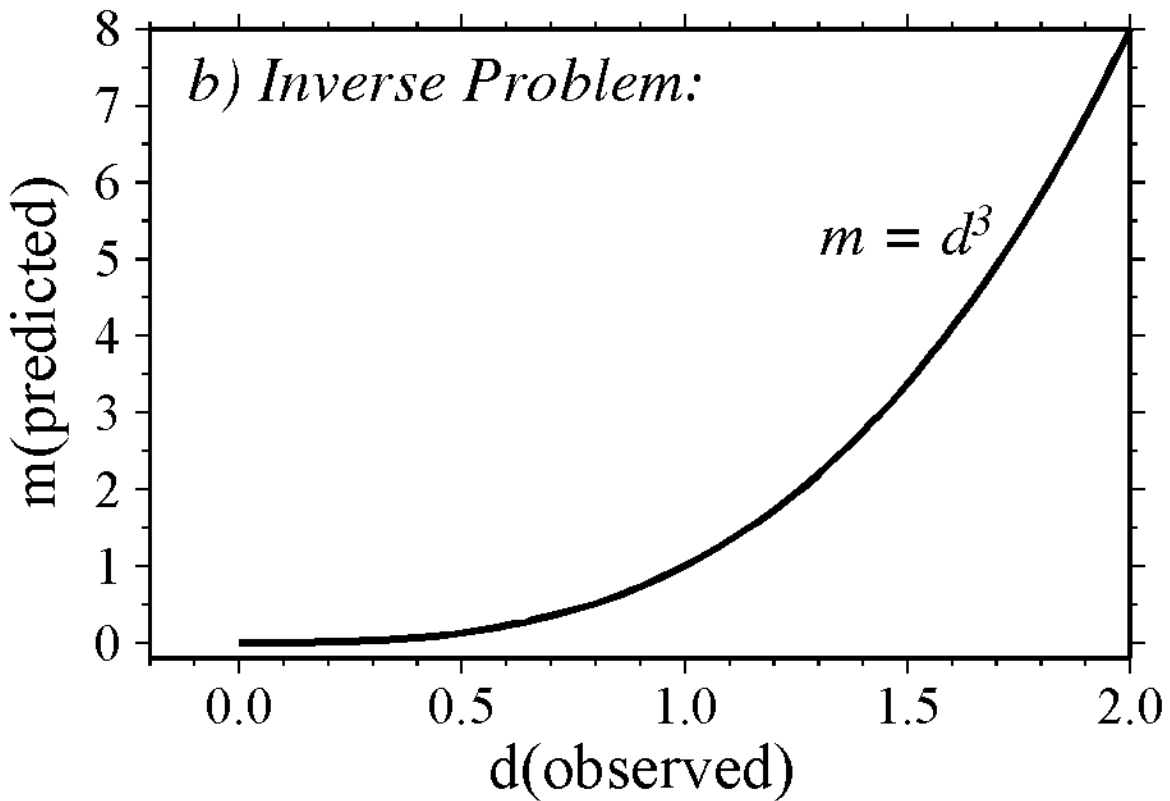
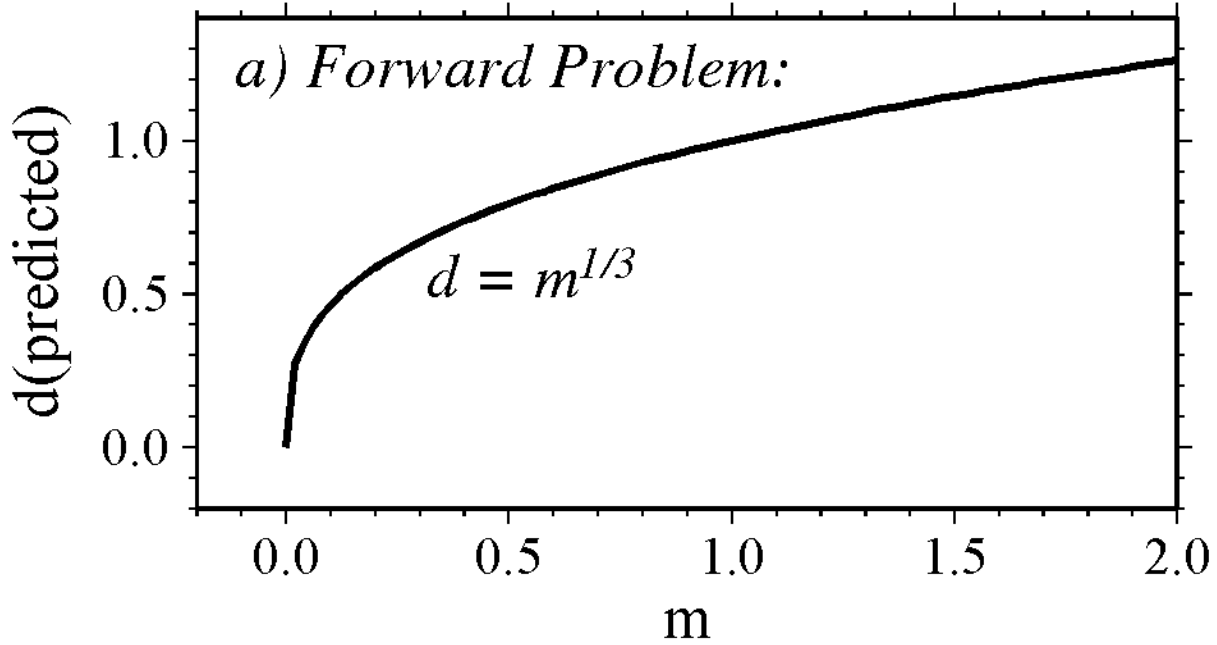
Suppose we consider a case where $d^{\text{true}} = 1$. The true solution is $m = 1$. A generalized inverse analysis leads to a linearized estimate of the uncertainty in the solution, [cov_u \mathbf{m}], of 9. This analysis assumes Gaussian noise with mean zero and variance 1. If we use Gaussian noise with mean zero and standard deviation $\sigma = 0.25$ (i.e., variance = 0.0625) then [cov \mathbf{m}] = 0.5625. The simple nature of this problem leads to an amplification by a factor of 9 between the data variance and the linearized estimate of the solution variance.

Now consider an experiment in which 50,000 noisy data values are collected. The noise in these data has a mean of 0.0 and a standard deviation of 0.25. For each noisy data value a solution is found from the above equations. This will create 50,000 estimates of the solution. Distributions of both the data noise and the solution are shown in the figures after those for the forward and inverse problems in Equations (7.111)–(7.112).

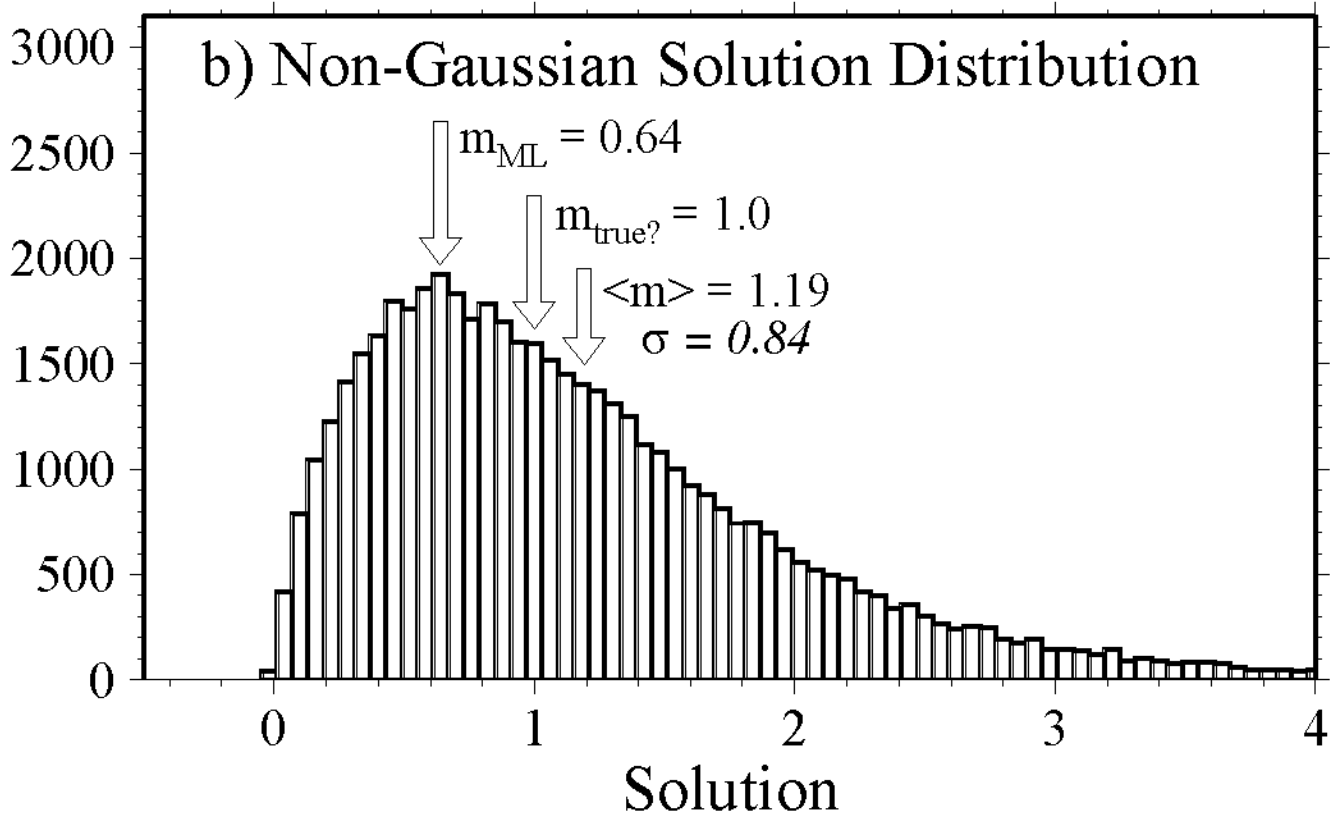
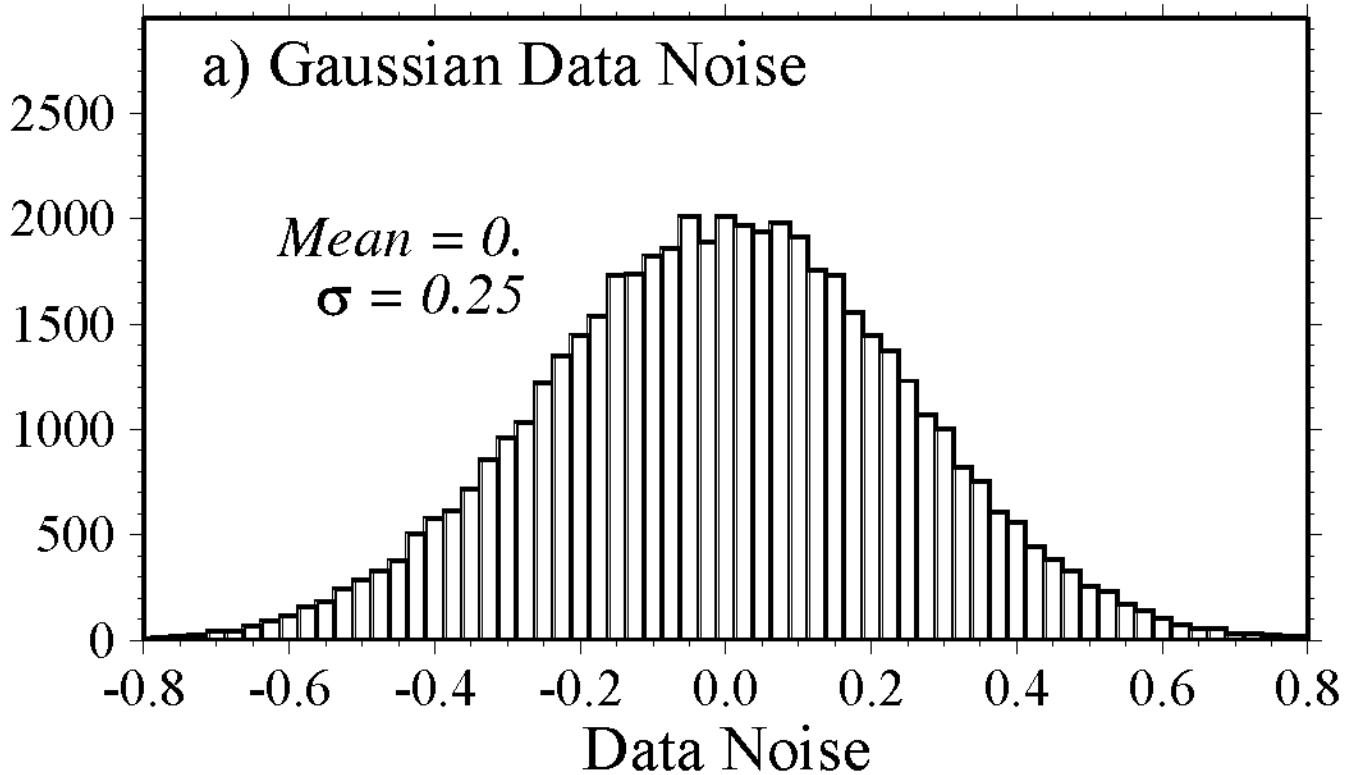
Note that due to the nonlinear nature of the forward problem, the distribution of solutions is not even Gaussian. The mean value, $\langle m \rangle$, is 1.18, greater than the “true” value of 1. The standard deviation is 0.84. Also shown on the figure is the maximum likelihood solution \mathbf{m}_{ML} , as determined empirically from the distribution.

The purpose of this example is to show that caution must be applied to the interpretation of all inverse problems, but especially nonlinear ones.

Forward and Inverse Problems



Mapping Noisy Data: 50,000 Experiments



CHAPTER 8: VARIATIONS OF THE GENERALIZED INVERSE

8.1 Linear Transformations

8.1.1 Analysis of the Generalized Inverse Operator \mathbf{G}_g^{-1}

Recall Equation (2.21)

$$\mathbf{ABC} = \mathbf{D} \quad (2.21)$$

which states that if the matrix \mathbf{D} is given by the product of matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} , then each column of \mathbf{D} is the weighted sum of the columns of \mathbf{A} and each row of \mathbf{D} is the weighted sum of the rows of \mathbf{C} . Applying this to $\mathbf{Gm} = \mathbf{d}$, we saw in Equation (2.15) that the data vector \mathbf{d} is the weighted sum of the columns of \mathbf{G} . Note that both the data vector and the columns of \mathbf{G} are $N \times 1$ vectors in data space.

We can extend this analysis by using singular-value decomposition. Specifically, writing out \mathbf{G} as

$$\mathbf{G} = \mathbf{U}_P \quad \Lambda_P \quad \mathbf{V}_P^T \quad (6.65)$$

$$N \times M \quad N \times P \quad P \times P \quad P \times M$$

Each column of \mathbf{G} is now seen as a weighted sum of the columns of \mathbf{U}_P . Each column of \mathbf{G} is an $N \times 1$ dimensional vector (i.e., in data space), and is the weighted sum of the P eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_P$ in \mathbf{U}_P . Each row of \mathbf{G} is a weighted sum of the rows of \mathbf{V}_P^T , or equivalently, the columns of \mathbf{V}_P . Each row of \mathbf{G} is a $1 \times M$ row vector in model space. It is the weighted sum of the P eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_P$ in \mathbf{V}_P .

A similar analysis may be considered for the generalized inverse operator, where

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \quad \Lambda_P^{-1} \quad \mathbf{U}_P^T \quad (7.9)$$

$$M \times N \quad M \times P \quad P \times P \quad P \times N$$

Each column of \mathbf{G}_g^{-1} is a weighted sum of the columns of \mathbf{V}_P . Each row of \mathbf{G}_g^{-1} is the weighted sum of the rows of \mathbf{U}_P^T , or equivalently, the columns of \mathbf{U}_P .

Let us now consider what happens in the system of equations $\mathbf{Gm} = \mathbf{d}$ when we take one of the eigenvectors in \mathbf{V}_P as \mathbf{m} . Let $\mathbf{m} = \mathbf{v}_i$, the i th eigenvector in \mathbf{V}_P . Then

$$\mathbf{Gv}_i = \mathbf{U}_P \quad \Lambda_P \quad \mathbf{V}_P^T \quad \mathbf{v}_i \quad (8.1)$$

$$N \times 1 \quad N \times P \quad P \times P \quad P \times M \quad M \times 1$$

We can expand this as

$$\mathbf{G}\mathbf{v}_i = \mathbf{U}_P \Lambda_P \begin{bmatrix} \cdots & v_1^T & \cdots \\ & \vdots & \\ \cdots & v_i^T & \cdots \\ & \vdots & \\ \cdots & v_P^T & \cdots \end{bmatrix} \mathbf{v}_i \quad (8.2)$$

The product of \mathbf{V}_P^T with \mathbf{v}_i is a $P \times 1$ vector with zeros everywhere except for the i th row, which represents the dot product of \mathbf{v}_i with itself.

Continuing, we have

$$\begin{aligned} \mathbf{G}\mathbf{v}_i &= \mathbf{U}_P \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \lambda_i & 0 \\ 0 & \cdots & 0 & \lambda_P \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} u_{11} & \cdots & u_{1i} & \cdots & u_{1P} \\ u_{21} & & u_{2i} & & u_{2P} \\ \vdots & & \vdots & & \vdots \\ u_{N1} & \cdots & u_{Ni} & \cdots & u_{NP} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \lambda_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \lambda_i \begin{bmatrix} u_{1i} \\ u_{2i} \\ \vdots \\ u_{Ni} \end{bmatrix} \end{aligned} \quad (8.3)$$

Or simply,

$$\mathbf{G}\mathbf{v}_i = \lambda_i \mathbf{u}_i \quad (8.4)$$

This is, of course, simply the statement of the shifted eigenvalue problem from Equation (6.16). The point was not, however, to reinvent the shifted eigenvalue problem, but to emphasize the linear algebra, or mapping, between vectors in model and data space.

Note that \mathbf{v}_i , a unit-length vector in model space, is transformed into a vector of length λ_i (since \mathbf{u}_i is also of unit length) in data space. If λ_i is large, then a unit-length change in model

space in the \mathbf{v}_i direction will have a large effect on the data. Conversely, if λ_i is small, then a unit length change in model space in the \mathbf{v}_i direction will have little effect on the data.

8.1.2 \mathbf{G}_g^{-1} Operating on a Data Vector \mathbf{d}

Now consider a similar analysis for the generalized inverse operator \mathbf{G}_g^{-1} , which operates on a data vector \mathbf{d} . Suppose that \mathbf{d} is given by one of the eigenvectors in \mathbf{U}_P , say \mathbf{u}_i . Then

$$\mathbf{G}_g^{-1} \mathbf{u}_i = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \mathbf{u}_i \quad (8.5)$$

$M \times 1 \quad M \times P \quad P \times P \quad P \times N \quad N \times 1$

Following the development above, note that the product of \mathbf{U}_P^T with \mathbf{u}_i is a $P \times 1$ vector with zeros everywhere except the i th row, which represents the dot product of \mathbf{u}_i with itself. Then

$$\mathbf{G}_g^{-1} \mathbf{u}_i = \mathbf{V}_P \begin{bmatrix} \lambda_1^{-1} & 0 & \dots & 0 \\ 0 & \ddots & & \\ \vdots & & \lambda_i^{-1} & \\ 0 & \dots & 0 & \lambda_P^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Continuing,

$$\begin{aligned} &= \begin{bmatrix} v_{11} & \dots & v_{1i} & \dots & v_{1P} \\ v_{21} & & v_{2i} & & v_{2P} \\ \vdots & & \vdots & & \vdots \\ v_{M1} & \dots & v_{Mi} & \dots & v_{MP} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \lambda_i^{-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \lambda_i^{-1} \begin{bmatrix} v_{1i} \\ v_{2i} \\ \vdots \\ v_{Mi} \end{bmatrix} \end{aligned} \quad (8.6a)$$

Or simply,

$$\mathbf{G}_g^{-1} \mathbf{u}_i = \lambda_i^{-1} \mathbf{v}_i \quad (8.6b)$$

This is not a statement of the shifted eigenvalue problem, but has an important implication for the mapping between data and model spaces. Specifically, it implies that a unit-length vector (\mathbf{u}_i) in data space is transformed into a vector of length $1/\lambda_i$ in model space. If λ_i is large, then small changes in \mathbf{d} in the direction of \mathbf{u}_i will have little effect in model space. This is good if, as usual, these small changes in the data vector are associated with noise. If λ_i is small, however, then small changes in \mathbf{d} in the \mathbf{u}_i direction will have a large effect on the model parameter estimates. This reflects a basic instability in inverse problems whenever there are small, nonzero singular values. Noise in the data, in directions parallel to eigenvectors associated with small singular values, will be amplified into very unstable model parameter estimates.

Note also that there is an intrinsic relationship, or coupling, between the eigenvectors \mathbf{v}_i in model space and \mathbf{u}_i in data space. When \mathbf{G} operates on \mathbf{v}_i , it returns \mathbf{u}_i , scaled by the singular value λ_i . Conversely, when \mathbf{G}_g^{-1} operates on \mathbf{u}_i it returns \mathbf{v}_i , scaled by λ_i^{-1} . This represents a very strong coupling between \mathbf{v}_i and \mathbf{u}_i directions, even though the former are in model space and the latter are in data space. Finally, the linkage between these vectors depends very strongly on the size of the nonzero singular value λ_i .

8.1.3 Mapping Between Data and Model Space: An Example

One useful way to graphically represent the mapping back and forth between model and data spaces is with the use of “stick figures.” These are formed by plotting the components of the eigenvectors in model and data space for each model parameter and observation as a “stick,” or line, whose length is given by the size of the component. These can be very helpful in illustrating directions in model space associated with stability and instability, as well as directions in data space where noise will have a large effect on the estimated solution.

For example, recall the previous example, given by

$$\begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.01 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 2.00 \\ 4.10 \end{bmatrix} \quad (7.79)$$

The singular values and associated eigenvectors are given by

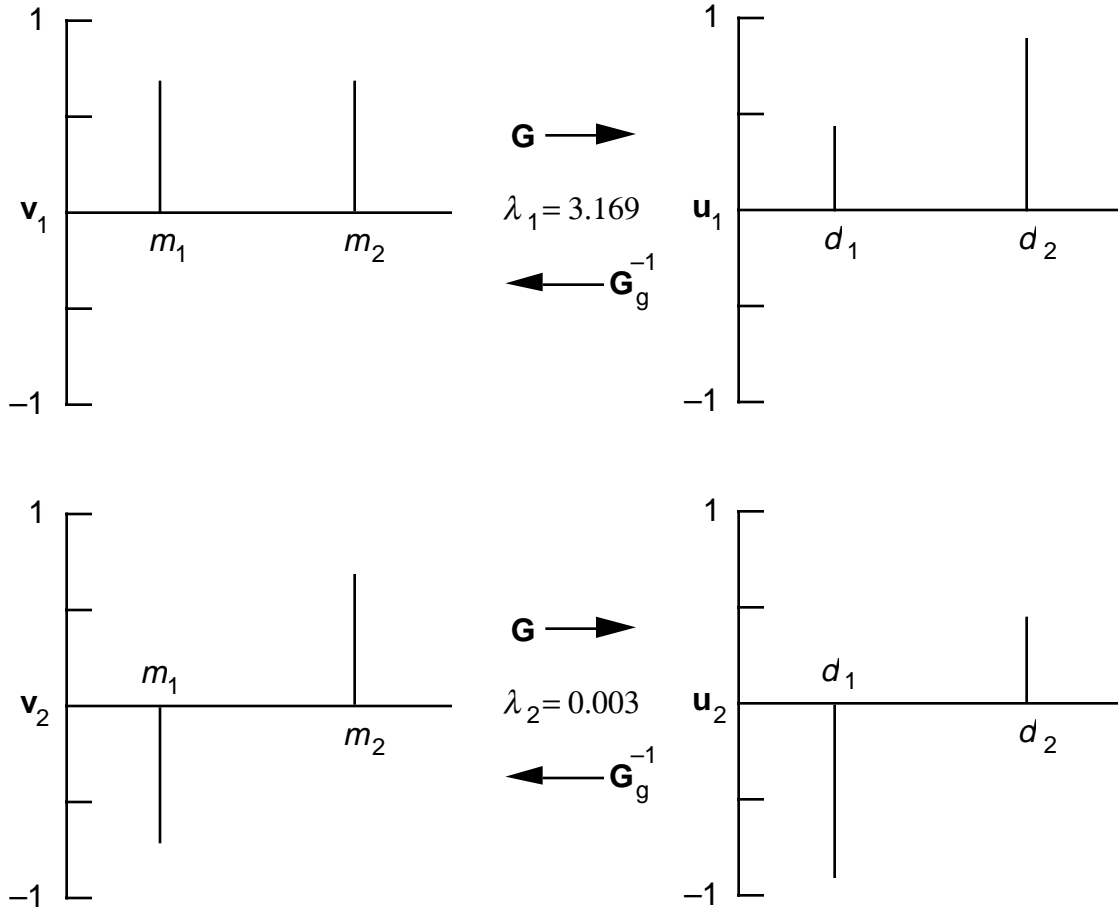
$$\lambda_1 = 3.169 \text{ and } \lambda_2 = 0.00316$$

$$\mathbf{V}_P = \mathbf{V} = \begin{bmatrix} 0.706 & -0.710 \\ 0.709 & 0.704 \end{bmatrix}$$

and

$$\mathbf{U}_P = \mathbf{U} = \begin{bmatrix} 0.446 & -0.895 \\ 0.895 & 0.446 \end{bmatrix} \quad (8.7)$$

From this information, we may plot the following figure:



From \mathbf{V}_P , we see that $\mathbf{v}_1 = [0.706, 0.709]^T$. Thus, on the figure for \mathbf{v}_1 , the component along m_1 is +0.706, while the component along m_2 is +0.709. Similarly, $\mathbf{u}_1 = [0.446, 0.895]^T$, and thus the components along d_1 and d_2 are +0.446 and +0.895, respectively. For $\mathbf{v}_2 = [-0.710, 0.704]^T$, the components along m_1 and m_2 are -0.710 and +0.704, respectively. Finally, the components of $\mathbf{u}_2 = [-0.895, 0.446]^T$ along d_1 and d_2 are -0.895 and 0.446, respectively.

These figures illustrate, in a simple way, the mapping back and forth between model and data space. For example, the top figure shows that a unit length change in model space in the $[0.706, 0.709]^T$ direction will be mapped by \mathbf{G} into a change of length 3.169 in data space in the $[0.446, 0.895]^T$ direction. A unit length change in data space along the $[0.446, 0.895]^T$ direction will be mapped by \mathbf{G}^{-1} into a change of length $1/(3.169) = 0.316$ in model space in the $[0.706, 0.709]^T$ direction. This is a stable mapping back and forth, since noise in the data is damped when it is mapped back into model space. The pairing between \mathbf{v}_2 and \mathbf{u}_2 directions is less stable, however, since a unit length change in data space parallel to \mathbf{u}_2 will be mapped back into a change of length $1/(0.00316) = 317$ parallel to \mathbf{v}_2 . The \mathbf{v}_2 direction in model space will be associated with a very large variance. Since \mathbf{v}_2 has significant components along both m_1 and m_2 , they will both individually have large variances, as seen in the unit model covariance matrix for this example, given by

$$[\text{cov}_{\mathbf{u}} \mathbf{m}] = \begin{bmatrix} 50,551 & -50,154 \\ -50,154 & 49,753 \end{bmatrix} \quad (8.8)$$

For a particular inverse problem, these figures can help one understand both the directions in model space that affect the data the least or most and the directions in data space along which noise will affect the estimated solution the least or most.

8.2 Including Prior Information, or the Weighted Generalized Inverse

8.2.1 Mathematical Background

As we have seen, the generalized inverse operator is a very powerful operator, combining the attributes of both least squares and minimum length estimators. Specifically, the generalized inverse minimizes both

$$\mathbf{e}^T \mathbf{e} = [\mathbf{d} - \mathbf{d}^{\text{pre}}]^T [\mathbf{d} - \mathbf{d}^{\text{pre}}] = [\mathbf{d} - \mathbf{G}\mathbf{m}]^T [\mathbf{d} - \mathbf{G}\mathbf{m}] \quad (8.9)$$

and $[\mathbf{m} - \langle \mathbf{m} \rangle]^T [\mathbf{m} - \langle \mathbf{m} \rangle]$, where $\langle \mathbf{m} \rangle$ is the *a priori* estimate of the solution.

As discussed in Chapter 3, however, it is useful to include as much prior information into an inverse problem as possible. Two forms of prior information were included in weighted least squares and weighted minimum length, and resulted in new minimization criteria given by

$$\mathbf{e}^T \mathbf{W}_e \mathbf{e} = \mathbf{e}^T [\text{cov } \mathbf{d}]^{-1} \mathbf{e} = [\mathbf{d} - \mathbf{G}\mathbf{m}]^T [\text{cov } \mathbf{d}]^{-1} [\mathbf{d} - \mathbf{G}\mathbf{m}] \quad (8.10)$$

and

$$\mathbf{m}^T \mathbf{W}_m \mathbf{m} = [\mathbf{m} - \langle \mathbf{m} \rangle]^T [\text{cov } \mathbf{m}]^{-1} [\mathbf{m} - \langle \mathbf{m} \rangle] \quad (8.11)$$

where $[\text{cov } \mathbf{d}]$ and $[\text{cov } \mathbf{m}]$ are *a priori* data and model covariance matrices, respectively. It is possible to include this information in a generalized inverse analysis as well.

The basic procedure is as follows. First, transform the problem into a coordinate system where the new data and model parameters each have uncorrelated errors and unit variance. The transformations are based on the information contained in the *a priori* data and model parameter covariance matrices. Then, perform a generalized inverse analysis in the transformed coordinate system. This is the appropriate inverse operator because both of the covariance matrices are identity matrices. Finally, transform everything back to the original coordinates to obtain the final solution.

One may assume that the data covariance matrix $[\text{cov } \mathbf{d}]$ is a positive definite Hermitian matrix. This is equivalent to assuming that all variances are positive, and none of the correlation coefficients are exactly equal to plus or minus one. Then the data covariance matrix can be decomposed as

$$[\text{cov } \mathbf{d}] = \begin{matrix} \mathbf{B} & \Lambda_d & \mathbf{B}^T \\ N \times N & N \times N & N \times N \end{matrix} \quad (8.12)$$

where Λ_d is a diagonal matrix containing the eigenvalues of $[\text{cov } \mathbf{d}]$ and \mathbf{B} is an orthonormal matrix containing the associated eigenvectors. \mathbf{B} is orthonormal because $[\text{cov } \mathbf{d}]$ is Hermitian, and all of the eigenvalues are positive because $[\text{cov } \mathbf{d}]$ is positive definite.

The inverse data covariance matrix is easily found as

$$[\text{cov } \mathbf{d}]^{-1} = \begin{matrix} \mathbf{B} & \Lambda_d^{-1} & \mathbf{B}^T \\ N \times N & N \times N & N \times N \end{matrix} \quad (8.13)$$

where we have taken advantage of the fact that \mathbf{B} is an orthonormal matrix. It is convenient to write the right-hand side of (8.13) as

$$\begin{matrix} \mathbf{B} & \Lambda_d^{-1} & \mathbf{B}^T \\ N \times N & N \times N & N \times N \end{matrix} = \begin{matrix} \mathbf{D}^T & \mathbf{D} \\ N \times N & N \times N \end{matrix} \quad (8.14)$$

where

$$\mathbf{D} = \Lambda_d^{-1/2} \mathbf{B}^T \quad (8.15)$$

Thus,

$$[\text{cov } \mathbf{d}]^{-1} = \begin{matrix} \mathbf{D}^T & \mathbf{D} \\ N \times N & N \times N \end{matrix} \quad (8.16)$$

The reason for writing the data covariance matrix in terms of \mathbf{D} will be clear when we introduce the transformed data vector. The covariance matrix itself can be expressed in terms of \mathbf{D} as

$$\begin{aligned} [\text{cov } \mathbf{d}] &= \{[\text{cov } \mathbf{d}]^{-1}\}^{-1} \\ &= [\mathbf{D}^T \mathbf{D}]^{-1} \\ &= \mathbf{D}^{-1} [\mathbf{D}^T]^{-1} \end{aligned} \quad (8.17)$$

Similarly, the positive definite Hermitian model covariance matrix may be decomposed as

$$[\text{cov } \mathbf{m}] = \begin{matrix} \mathbf{M} & \Lambda_m & \mathbf{M}^T \\ M \times M & M \times M & M \times M \end{matrix} \quad (8.18)$$

where Λ_m is a diagonal matrix containing the eigenvalues of $[\text{cov } \mathbf{m}]$ and \mathbf{M} is an orthonormal matrix containing the associated eigenvectors.

The inverse model covariance matrix is thus given by

$$[\text{cov } \mathbf{m}]^{-1} = \begin{matrix} \mathbf{M} & \Lambda_m^{-1} & \mathbf{M}^T \\ M \times M & M \times M & M \times M \end{matrix} \quad (8.19)$$

where, as before, we have taken advantage of the fact that \mathbf{M} is an orthonormal matrix. The right-hand side of (8.18) can be written as

$$\begin{matrix} \mathbf{M} & \Lambda_m^{-1} & \mathbf{M}^T \\ M \times M & M \times M & M \times M \end{matrix} = \begin{matrix} \mathbf{S}^T & \mathbf{S} \\ M \times M & M \times M \end{matrix} \quad (8.20)$$

where

$$\mathbf{S} = \Lambda_m^{-1/2} \mathbf{M}^T \quad (8.21)$$

Thus,

$$[\text{cov } \mathbf{m}]^{-1} = \begin{matrix} \mathbf{S}^T & \mathbf{S} \\ M \times M & M \times M \end{matrix} \quad (8.22)$$

As before, it is possible to write the covariance matrix in terms of \mathbf{S} as

$$\begin{aligned} [\text{cov } \mathbf{m}] &= \{[\text{cov } \mathbf{m}]^{-1}\}^{-1} \\ &= [\mathbf{S}^T \mathbf{S}]^{-1} \\ &= \mathbf{S}^{-1} [\mathbf{S}^T]^{-1} \end{aligned} \quad (8.23)$$

8.2.2 Coordinate System Transformation of Data and Model Parameter Vectors

The utility of \mathbf{D} and \mathbf{S} can now be seen as we introduce transformed data and model parameter vectors. First, we introduce a transformed data vector \mathbf{d}' as

$$\mathbf{d}' = \Lambda_d^{-1/2} \mathbf{B}^T \mathbf{d} \quad (8.24)$$

or

$$\mathbf{d}' = \mathbf{D} \mathbf{d} \quad (8.25)$$

The transformed model parameter \mathbf{m}' is given by

$$\mathbf{m}' = \Lambda_m^{-1/2} \mathbf{M}^T \mathbf{m} \quad (8.26)$$

or

$$\mathbf{m}' = \mathbf{S} \mathbf{m} \quad (8.27)$$

The forward operator \mathbf{G} must also be transformed into \mathbf{G}' , the new coordinates. The transformation can be found by recognizing that

$$\mathbf{G}' \mathbf{m}' = \mathbf{d}' \quad (8.28)$$

$$\mathbf{G}' \mathbf{S} \mathbf{m} = \mathbf{D} \mathbf{d} \quad (8.29)$$

or

$$\mathbf{D}^{-1} \mathbf{G}' \mathbf{S} \mathbf{m} = \mathbf{d} = \mathbf{G} \mathbf{m} \quad (8.30)$$

That is

$$\mathbf{D}^{-1}\mathbf{G}'\mathbf{S} = \mathbf{G} \quad (8.31)$$

Finally, by pre- and postmultiplying by \mathbf{D} and \mathbf{S}^{-1} , respectively, we obtain \mathbf{G}' as

$$\mathbf{G}' = \mathbf{D}\mathbf{G}\mathbf{S}^{-1} \quad (8.32)$$

The transformations back from the primed coordinates to the original coordinates are given by

$$\mathbf{d} = \mathbf{B}\Lambda_d^{1/2} \mathbf{d}' \quad (8.33)$$

or

$$\mathbf{d} = \mathbf{D}^{-1}\mathbf{d}' \quad (8.34)$$

$$\mathbf{m} = \mathbf{M}\Lambda_m^{1/2} \mathbf{m}' \quad (8.35)$$

or

$$\mathbf{m} = \mathbf{S}^{-1}\mathbf{m}' \quad (8.36)$$

and

$$\mathbf{G} = \mathbf{B}\Lambda_d^{1/2} \mathbf{G}'\Lambda_m^{1/2} \mathbf{M}^T \quad (8.37)$$

or

$$\mathbf{G} = \mathbf{D}^{-1}\mathbf{G}'\mathbf{S} \quad (8.38)$$

In the new coordinate system, the generalized inverse will minimize

$$\mathbf{e}'^T\mathbf{e}' = [\mathbf{d}' - \mathbf{d}'^{\text{pre}}]^T[\mathbf{d}' - \mathbf{d}'^{\text{pre}}] = [\mathbf{d}' - \mathbf{G}'\mathbf{m}']^T[\mathbf{d}' - \mathbf{G}'\mathbf{m}'] \quad (8.39)$$

and $[\mathbf{m}']^T\mathbf{m}'$.

Replacing \mathbf{d}' , \mathbf{m}' and \mathbf{G}' in (8.39) with Equations (8.24)–(8.32), we have

$$\begin{aligned} [\mathbf{d}' - \mathbf{G}'\mathbf{m}']^T[\mathbf{d}' - \mathbf{G}'\mathbf{m}'] &= [\mathbf{D}\mathbf{d} - \mathbf{D}\mathbf{G}\mathbf{S}^{-1}\mathbf{S}\mathbf{m}]^T[\mathbf{D}\mathbf{d} - \mathbf{D}\mathbf{G}\mathbf{S}^{-1}\mathbf{S}\mathbf{m}] \\ &= [\mathbf{D}\mathbf{d} - \mathbf{D}\mathbf{G}\mathbf{m}]^T[\mathbf{D}\mathbf{d} - \mathbf{D}\mathbf{G}\mathbf{m}] \\ &= \{\mathbf{D}[\mathbf{d} - \mathbf{G}\mathbf{m}]\}^T\{\mathbf{D}[\mathbf{d} - \mathbf{G}\mathbf{m}]\} \\ &= [\mathbf{d} - \mathbf{G}\mathbf{m}]^T\mathbf{D}^T\mathbf{D}[\mathbf{d} - \mathbf{G}\mathbf{m}] \\ &= [\mathbf{d} - \mathbf{G}\mathbf{m}]^T[\text{cov } \mathbf{d}]^{-1}[\mathbf{d} - \mathbf{G}\mathbf{m}] \end{aligned} \quad (8.40)$$

where we have used (8.16) to replace $\mathbf{D}^T\mathbf{D}$ with $[\text{cov } \mathbf{d}]^{-1}$.

Equation (8.40) shows that the unweighted misfit in the primed coordinate system is precisely the weighted misfit to be minimized in the original coordinates. Thus, the least squares solution in the primed coordinate system is equivalent to weighted least squares in the original coordinates.

Furthermore, using (8.27) for \mathbf{m}' , we have

$$\begin{aligned}\mathbf{m}'^T \mathbf{m}' &= [\mathbf{S}\mathbf{m}]^T \mathbf{S}\mathbf{m} \\ &= \mathbf{m}^T \mathbf{S}^T \mathbf{S}\mathbf{m} \\ &= \mathbf{m}^T [\text{cov } \mathbf{m}]^{-1} \mathbf{m}\end{aligned}\tag{8.41}$$

where we have used (8.22) to replace $\mathbf{S}^T \mathbf{S}$ with $[\text{cov } \mathbf{m}]^{-1}$.

Equation (8.41) shows that the unweighted minimum length solution in the new coordinate system is equivalent to the weighted minimum length solution in the original coordinate system. Thus minimum length in the new coordinate system is equivalent to weighted minimum length in the original coordinates.

8.2.3 The Maximum Likelihood Inverse Operator, Resolution, and Model Covariance

The generalized inverse operator in the primed coordinates can be transformed into an operator in the original coordinates. We will show later that this is, in fact, the maximum likelihood operator in the case where all distributions are Gaussian. Let this inverse operator be $\mathbf{G}_{\text{MX}}^{-1}$, and be given by

$$\begin{aligned}\mathbf{G}_{\text{MX}}^{-1} &= [\mathbf{D}^{-1} \mathbf{G}' \mathbf{S}]_g^{-1} \\ &= \mathbf{S}^{-1} [\mathbf{G}']_g^{-1} \mathbf{D}\end{aligned}\tag{8.42}$$

The solution in the original coordinates, \mathbf{m}_{MX} , can be expressed either as

$$\mathbf{m}_{\text{MX}} = \mathbf{G}_{\text{MX}}^{-1} \mathbf{d}\tag{8.43}$$

or as

$$\begin{aligned}\mathbf{m}_{\text{MX}} &= \mathbf{S}^{-1} \mathbf{m}'_g \\ &= \mathbf{S}^{-1} [\mathbf{G}']_g^{-1} \mathbf{d}'\end{aligned}\tag{8.44}$$

Now that the operator has been expressed in the original coordinates, it is possible to calculate the resolution matrices and an *a posteriori* model covariance matrix.

The model resolution matrix \mathbf{R} is given by

$$\mathbf{R} = \mathbf{G}_{\text{MX}}^{-1} \mathbf{G}$$

$$\begin{aligned}
 &= \{\mathbf{S}^{-1}[\mathbf{G}'_g]^{-1} \mathbf{D}\} \{\mathbf{D}^{-1} \mathbf{G}' \mathbf{S}\} \\
 &= \mathbf{S}^{-1}[\mathbf{G}'_g]^{-1} \mathbf{G}' \mathbf{S} \\
 &= \mathbf{S}^{-1} \mathbf{R}' \mathbf{S}
 \end{aligned} \tag{8.45}$$

where \mathbf{R}' is the model resolution matrix in the transformed coordinate system.

Similarly, the data resolution matrix \mathbf{N} is given by

$$\begin{aligned}
 \mathbf{N} &= \mathbf{G} \mathbf{G}_{\text{MX}}^{-1} \\
 &= \{\mathbf{D}^{-1} \mathbf{G}' \mathbf{S}\} \{\mathbf{S}^{-1}[\mathbf{G}'_g]^{-1} \mathbf{D}\} \\
 &= \mathbf{D}^{-1} \mathbf{G}' [\mathbf{G}'_g]^{-1} \mathbf{D} \\
 &= \mathbf{D}^{-1} \mathbf{N}' \mathbf{D}
 \end{aligned} \tag{8.46}$$

The *a posteriori* model covariance matrix $[\text{cov } \mathbf{m}]_P$ is given by

$$[\text{cov } \mathbf{m}]_P = \mathbf{G}_{\text{MX}}^{-1} [\text{cov } \mathbf{d}] [\mathbf{G}_{\text{MX}}^{-1}]^T \tag{8.47}$$

Replacing $[\text{cov } \mathbf{d}]$ in (8.47) with (8.17) gives

$$\begin{aligned}
 [\text{cov } \mathbf{m}]_P &= \mathbf{G}_{\text{MX}}^{-1} \mathbf{D}^{-1} [\mathbf{D}^T]^{-1} [\mathbf{G}_{\text{MX}}^{-1}]^T \\
 &= \{\mathbf{S}^{-1}[\mathbf{G}'_g]^{-1} \mathbf{D}\} \mathbf{D}^{-1} [\mathbf{D}^T]^{-1} \{\mathbf{S}^{-1}[\mathbf{G}'_g]^{-1} \mathbf{D}\}^T \\
 &= \mathbf{S}^{-1} [\mathbf{G}'_g]^{-1} \mathbf{D} \mathbf{D}^{-1} [\mathbf{D}^T]^{-1} \mathbf{D}^T \{[\mathbf{G}'_g]^{-1}\}^T [\mathbf{S}^{-1}]^T \\
 &= \mathbf{S}^{-1} [\mathbf{G}'_g]^{-1} \{[\mathbf{G}'_g]^{-1}\}^T [\mathbf{S}^{-1}]^T \\
 &= \mathbf{S}^{-1} [\text{cov}_u \mathbf{m}'] [\mathbf{S}^{-1}]^T
 \end{aligned} \tag{8.48}$$

That is, an *a posteriori* estimate of model parameter uncertainties can be obtained by transforming the unit model covariance matrix from the primed coordinates back to the original coordinates.

It is important to realize that the transformations introduced by \mathbf{D} and \mathbf{S} in (8.24)–(8.38) are not, in general, orthonormal. Thus,

$$\mathbf{d}' = \mathbf{D} \mathbf{d} \tag{8.25}$$

implies that the length of the transformed data vector \mathbf{d}' is, in general, not equal to the length of the original data vector \mathbf{d} . The function of \mathbf{D} is to transform the data space into one in which the data errors are uncorrelated and all observations have unit variance. If the original data errors are uncorrelated, the data covariance matrix will be diagonal and \mathbf{B} , from

$$[\text{cov } \mathbf{d}]^{-1} = \begin{matrix} \mathbf{B} & \Lambda_d^{-1} & \mathbf{B}^T \\ N \times N & N \times N & N \times N \end{matrix} \quad (8.13)$$

will be an identity matrix. Then \mathbf{D} , given by

$$\mathbf{D} = \Lambda_d^{-1/2} \mathbf{B}^T \quad (8.15)$$

will be a diagonal matrix given by $\Lambda_d^{-1/2}$. The transformed data \mathbf{d}' are then given by

$$\mathbf{d}' = \Lambda_d^{-1/2} \mathbf{d}$$

or

$$d'_i = d_i / \sigma_{di} \quad i = 1, N \quad (8.49)$$

where σ_{di} is the data standard deviation for the i th observation. If the original data errors are uncorrelated, then each transformed observation is given by the original observation, divided by its standard deviation. The transformation in this case can be thought of as leaving the direction of each axis in data space unchanged, but stretching or compressing each axis, depending on the standard deviation. To see this, consider a vector in data space representing the d_1 axis. That is,

$$\mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8.50)$$

This data vector is transformed into

$$\mathbf{d}' = \begin{bmatrix} 1/\sigma_{d1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8.51)$$

That is, the direction of the axis is unchanged, but the magnitude is changed by $1/\sigma_{d1}$. If the data errors are correlated, then the axes in data space are rotated (by \mathbf{B}^T), and then stretched or compressed.

Very similar arguments can be made about the role of \mathbf{S} in model space. That is, if the *a priori* model covariance matrix is diagonal, then the directions of the transformed axes in model space are the same as in the original coordinates (i.e., m_1, m_2, \dots, m_M), but the lengths are stretched or compressed by the appropriate model parameter standard deviations. If the errors are correlated, then the axes in model space are rotated (by \mathbf{M}^T) before they are stretched or compressed.

8.2.4 Effect on Model- and Data-Space Eigenvectors

This stretching and compressing of directions in data and model space affects the eigenvectors as well. Let $\hat{\mathbf{V}}$ be the set of vectors transformed back into the original coordinates from \mathbf{V}' , the set of model eigenvectors in the primed coordinates. Thus,

$$\hat{\mathbf{V}} = \mathbf{S}^{-1}\mathbf{V}' \quad (8.52)$$

For example, suppose that $[\text{cov } \mathbf{m}]$ is diagonal, then

$$\hat{\mathbf{V}} = \Lambda_m^{1/2} \mathbf{V}' \quad (8.53)$$

For $\hat{\mathbf{v}}_i$, the i th vector in $\hat{\mathbf{V}}$, this implies

$$\begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \\ \vdots \\ \hat{v}_M \end{bmatrix}_i = \begin{bmatrix} \sigma_{m1} v_1 \\ \sigma_{m2} v_2 \\ \vdots \\ \sigma_{mM} v_M \end{bmatrix}_i \quad (8.54)$$

Clearly, for a general diagonal $[\text{cov } \mathbf{m}]$, $\hat{\mathbf{v}}_i$ will no longer have unit length. This is true whether or not $[\text{cov } \mathbf{m}]$ is diagonal. Thus, in general, the vectors in $\hat{\mathbf{V}}$ are not unit length vectors. They can, of course, be normalized to unit length. Perhaps more importantly, however, the directions of the $\hat{\mathbf{v}}_i$ have been changed, and the vectors in $\hat{\mathbf{V}}$ are no longer perpendicular to each other. Thus, the vectors in $\hat{\mathbf{V}}$ cannot be thought of as orthonormal eigenvectors, even if they have been normalized to unit length.

These vectors still play an important role in the inverse analysis, however. Recall that the solution \mathbf{m}_{MX} is given by

$$\mathbf{m}_{\text{MX}} = \mathbf{G}_{\text{MX}}^{-1} \mathbf{d} \quad (8.43)$$

or as

$$\begin{aligned} \mathbf{m}_{\text{MX}} &= \mathbf{S}^{-1} \mathbf{m}_g' \\ &= \mathbf{S}^{-1} [\mathbf{G}']_g^{-1} \mathbf{d}' \end{aligned} \quad (8.44)$$

We can expand (8.44) as

$$\begin{aligned} \mathbf{m}_{\text{MX}} &= \mathbf{S}^{-1} \mathbf{V}' [\Lambda'_p]^{-1} [\mathbf{U}'_p]^T \mathbf{D} \mathbf{d} \\ &= \hat{\mathbf{V}}_p [\Lambda'_p]^{-1} [\mathbf{U}'_p]^T \mathbf{D} \mathbf{d} \end{aligned} \quad (8.55)$$

Recall that the solution \mathbf{m}_{MX} can be thought of as a linear combination of the columns of the first matrix in a product of several matrices [see Equations (2.15)–(2.21)]. This implies that the solution \mathbf{m}_{MX} consists of a linear combination of the columns of $\hat{\mathbf{V}}_p$. The solution is still a linear combination of the vectors in $\hat{\mathbf{V}}_p$, even if they have been normalized to unit length. Thus, $\hat{\mathbf{V}}_p$ still

plays a fundamental role in the inverse analysis.

It is important to realize that [cov \mathbf{m}] will only affect the solution if $P < M$. If $P = M$, then $\mathbf{V}'_P = \mathbf{V}'$, and \mathbf{V}'_P spans all of model space. $\hat{\mathbf{V}}_P$ will also span all of solution space. In this case, all of model space can be expressed as a linear combination of the vectors in $\hat{\mathbf{V}}_P$, even though they are not an orthonormal set of vectors. Thus, the same solution will be reached, regardless of the values in [cov \mathbf{m}]. If $P < M$, however, the mapping of vectors from the primed coordinates back to the original space can affect the part of solution space that is spanned by $\hat{\mathbf{V}}_P$. We will return to this point later with a specific example.

Very similar arguments can be made for the data eigenvectors as well. Let $\hat{\mathbf{U}}$ be the set of vectors obtained by transforming the data eigenvectors \mathbf{U}' in the primed coordinates back into the original coordinates. Then

$$\hat{\mathbf{U}} = \mathbf{D}^{-1}\mathbf{U}' \quad (8.56)$$

In general, the vectors in $\hat{\mathbf{U}}$ will not be either of unit length or perpendicular to each other.

The predicted data $\hat{\mathbf{d}}$ are given by

$$\begin{aligned} \hat{\mathbf{d}} &= \mathbf{G} \mathbf{m}_{MX} \\ &= \mathbf{D}^{-1}\mathbf{G}'\mathbf{S}\mathbf{m}_{MX} \\ &= \mathbf{D}^{-1}\mathbf{U}'_P \Lambda'_P \mathbf{V}'_P \mathbf{S}\mathbf{m}_{MX} \\ &= \hat{\mathbf{U}}_P \Lambda'_P \mathbf{V}'_P \mathbf{S}\mathbf{m}_{MX} \quad (8.57) \end{aligned}$$

Thus, the predicted data are a linear combination of the columns of $\hat{\mathbf{U}}_P$.

It is important to realize that the transformations introduced by [cov \mathbf{d}] will only affect the solution if $P < N$. If $P = N$, then $\mathbf{U}'_P = \mathbf{U}'$, and \mathbf{U}'_P spans all of data space. The matrix $\hat{\mathbf{U}}_P$ will also span all of data space. In this case, all of data space can be expressed as a linear combination of the vectors in $\hat{\mathbf{U}}_P$, even though they are not an orthonormal set of vectors. Thus, the same solution will be reached, regardless of the values in [cov \mathbf{d}]. If $P < N$, however, the mapping of vectors from the primed coordinates back to the original space can affect the part of solution space that is spanned by $\hat{\mathbf{U}}_P$. We are now in a position to consider a specific example.

8.2.5 An Example

Consider the following specific example of the form $\mathbf{G}\mathbf{m} = \mathbf{d}$, where \mathbf{G} and \mathbf{d} are given by

$$\mathbf{G} = \begin{bmatrix} 1.00 & 1.00 \\ 2.00 & 2.00 \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} 4.00 \\ 5.00 \end{bmatrix} \quad (8.58)$$

If we assume for the moment that the *a priori* data and model parameter covariance matrices are identity matrices and perform a generalized inverse analysis, we obtain

$$P = 1 < M = N = 2$$

$$\lambda_1 = 3.162$$

$$\mathbf{V} = \begin{bmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 0.447 & 0.894 \\ 0.894 & -0.447 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.500 & 0.500 \\ 0.500 & 0.500 \end{bmatrix}$$

$$\mathbf{N} = \begin{bmatrix} 0.200 & 0.400 \\ 0.400 & 0.800 \end{bmatrix}$$

$$\mathbf{m}_g = \begin{bmatrix} 1.400 \\ 1.400 \end{bmatrix}$$

$$\hat{\mathbf{d}} = \begin{bmatrix} 2.800 \\ 5.600 \end{bmatrix}$$

$$\mathbf{e}^T \mathbf{e} = \mathbf{e}^T [\text{cov } \mathbf{d}]^{-1} \mathbf{e} = 1.800 \quad (8.59)$$

The two rows (or columns) of \mathbf{G} are linearly dependent, and thus the number of nonzero singular values is one. Thus, the first column of \mathbf{V} (or \mathbf{U}) gives \mathbf{V}_P (or \mathbf{U}_P), while the second column gives \mathbf{V}_0 (or \mathbf{U}_0). The generalized inverse solution \mathbf{m}_g must lie in \mathbf{V}_P space, and is thus parallel to the $[0.707, 0.707]^T$ direction in model space. Similarly, the predicted data $\hat{\mathbf{d}}$ must lie in \mathbf{U}_P space, and is thus parallel to the $[0.447, 0.894]^T$ direction in data space. The model resolution matrix \mathbf{R} indicates that only the sum, equally weighted, of the model parameters m_1 and m_2 is resolved. Similarly, the data resolution matrix \mathbf{N} indicates that only the sum of d_1 and d_2 , with more weight on d_2 , is resolved, or important, in constraining the solution.

Now let us assume that the *a priori* data and model parameter covariance matrices are not equal to a constant times the identity matrix. Suppose

$$[\text{cov } \mathbf{d}] = \begin{bmatrix} 4.362 & -2.052 \\ -2.052 & 15.638 \end{bmatrix}$$

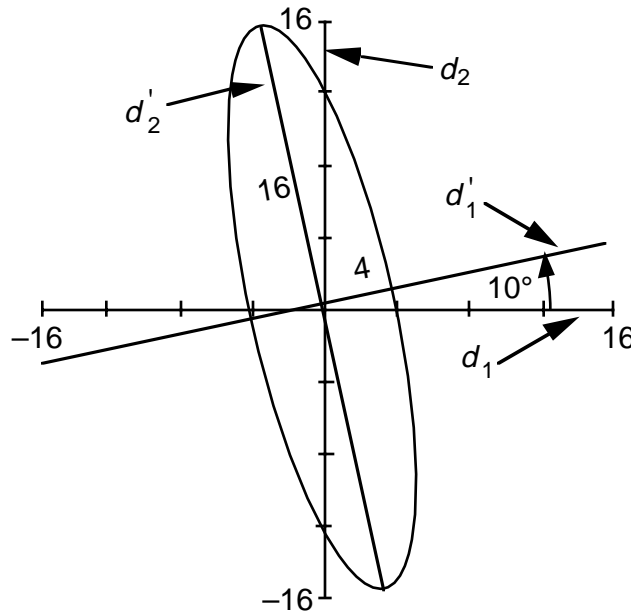
and

$$[\text{cov } \mathbf{m}] = \begin{bmatrix} 23.128 & 5.142 \\ 5.142 & 10.872 \end{bmatrix} \quad (8.60)$$

The data covariance matrix $[\text{cov } \mathbf{d}]$ can be decomposed as

$$\begin{aligned} [\text{cov } \mathbf{d}] &= \mathbf{B}\Lambda_d\mathbf{B}^T \\ &= \begin{bmatrix} 0.985 & -0.174 \\ 0.174 & 0.985 \end{bmatrix} \begin{bmatrix} 4.000 & 0.000 \\ 0.000 & 16.000 \end{bmatrix} \begin{bmatrix} 0.985 & 0.174 \\ -0.174 & 0.985 \end{bmatrix} \\ &= \begin{bmatrix} 4.362 & -2.052 \\ -2.052 & 15.638 \end{bmatrix} \end{aligned} \quad (8.61)$$

Recall that \mathbf{B} contains the eigenvectors of the symmetric matrix $[\text{cov } \mathbf{d}]$. Furthermore, these eigenvectors represent the directions of the major and minor axes of an ellipse. Thus, for the present case, the first vector in \mathbf{B} , $[0.985, 0.174]^T$, is the direction in data space of the minor axis of an ellipse having a half-length of 4. Similarly, the second vector in \mathbf{B} , $[-0.174, 0.985]^T$, is the direction in data space of the major axis of an ellipse having length 16. The eigenvectors in \mathbf{B}^T represent a 10° counterclockwise rotation of data space, as shown below:



The negative off-diagonal entries in $[\text{cov } \mathbf{d}]$ indicate a negative correlation of errors between d_1 and d_2 . Compare the figure above with figure (c) on page 23 of these notes.

The inverse data covariance $[\text{cov } \mathbf{d}]^{-1}$ can also be written as

$$[\text{cov } \mathbf{d}]^{-1} = \mathbf{B}\Lambda_d^{-1}\mathbf{B}^T$$

$$= \mathbf{D}^T \mathbf{D} \quad (8.62)$$

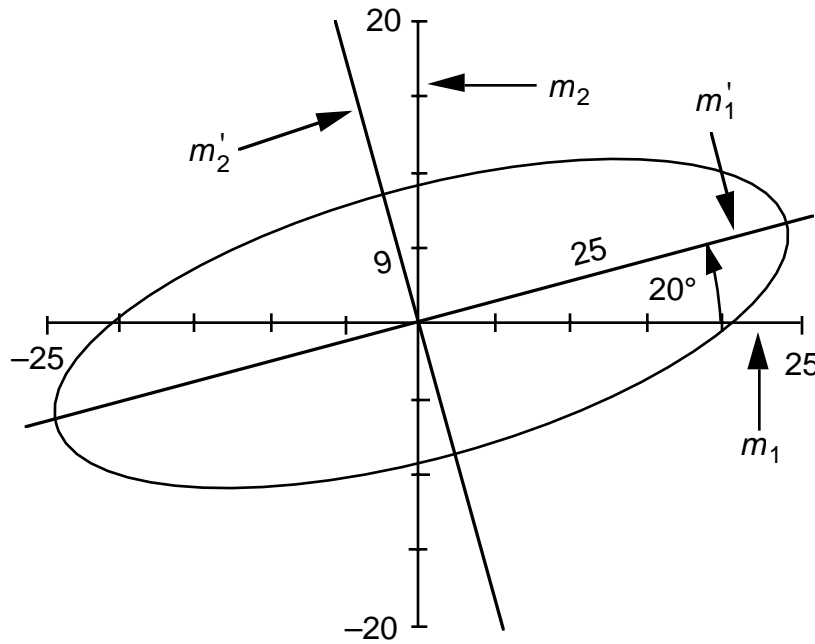
where \mathbf{D} is given by

$$\begin{aligned} \mathbf{D} &= \Lambda_d^{-1/2} \mathbf{B}^T \\ &= \begin{bmatrix} 0.492 & 0.087 \\ -0.043 & 0.246 \end{bmatrix} \end{aligned} \quad (8.63)$$

Similarly, the model covariance matrix $[\text{cov } \mathbf{m}]$ can be decomposed as

$$\begin{aligned} [\text{cov } \mathbf{m}] &= \mathbf{M} \Lambda_m \mathbf{M}^T \\ &= \begin{bmatrix} 0.940 & -0.342 \\ 0.342 & 0.940 \end{bmatrix} \begin{bmatrix} 25.000 & 0.000 \\ 0.000 & 9.000 \end{bmatrix} \begin{bmatrix} 0.940 & 0.342 \\ -0.342 & 0.940 \end{bmatrix} \end{aligned} \quad (8.64)$$

The matrix \mathbf{M}^T represents a 20° counterclockwise rotation of the m_1 and m_2 axes in model space. In the new coordinate system, the *a priori* model parameter errors are uncorrelated and have variances of 25 and 9, respectively. The major and minor axes of the error ellipse are along the $[0.940, 0.342]^T$ and $[-0.342, 0.940]^T$ directions, respectively. The geometry of the problem in model space is shown below:



The inverse model parameter covariance matrix $[\text{cov } \mathbf{m}]^{-1}$ can also be written as

$$\begin{aligned} [\text{cov } \mathbf{m}]^{-1} &= \mathbf{M} \Lambda_m^{-1} \mathbf{M}^T \\ &= \mathbf{S}^T \mathbf{S} \end{aligned} \quad (8.65)$$

where \mathbf{S} is given by

$$\begin{aligned} \mathbf{S} &= \Lambda_m^{-1/2} \mathbf{M}^T \\ &= \begin{bmatrix} 0.188 & 0.068 \\ -0.114 & 0.313 \end{bmatrix} \end{aligned} \quad (8.66)$$

With the information in \mathbf{D} and \mathbf{S} , it is now possible to transform \mathbf{G} , \mathbf{d} , and \mathbf{m} into \mathbf{G}' , \mathbf{d}' , and \mathbf{m}' in the new coordinate system:

$$\begin{aligned} \mathbf{G}' &= \mathbf{DGS}^{-1} \\ &= \begin{bmatrix} 0.492 & 0.087 \\ -0.043 & 0.246 \end{bmatrix} \begin{bmatrix} 1.000 & 1.000 \\ 2.000 & 2.000 \end{bmatrix} \begin{bmatrix} 4.698 & -1.026 \\ 1.710 & 2.819 \end{bmatrix} \\ &= \begin{bmatrix} 4.26844 & 1.19424 \\ 2.87739 & 0.80505 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{d}' &= \mathbf{Dd} \\ &= \begin{bmatrix} 0.492 & 0.087 \\ -0.043 & 0.246 \end{bmatrix} \begin{bmatrix} 4.000 \\ 5.000 \end{bmatrix} \\ &= \begin{bmatrix} 2.40374 \\ 1.05736 \end{bmatrix} \end{aligned} \quad (8.67)$$

In the new coordinate system, the data and model parameter covariance matrices are identity matrices. Thus, a generalized inverse analysis gives

$$P = 1 < M = N = 2$$

$$\lambda_1 = 5.345$$

$$\mathbf{V}' = \begin{bmatrix} 0.963 & -0.269 \\ 0.269 & 0.963 \end{bmatrix}$$

$$\mathbf{U}' = \begin{bmatrix} 0.829 & -0.559 \\ 0.559 & 0.829 \end{bmatrix}$$

$$\mathbf{G}'_g^{-1} = \begin{bmatrix} 0.149 & 0.101 \\ 0.042 & 0.028 \end{bmatrix}$$

$$\mathbf{R}' = \begin{bmatrix} 0.927 & 0.259 \\ 0.259 & 0.073 \end{bmatrix}$$

$$\mathbf{N}' = \begin{bmatrix} 0.688 & 0.463 \\ 0.463 & 0.312 \end{bmatrix}$$

$$\mathbf{m}'_g = \begin{bmatrix} 0.466 \\ 0.130 \end{bmatrix}$$

$$\hat{\mathbf{d}}' = \begin{bmatrix} 2.143 \\ 1.145 \end{bmatrix}$$

$$[\mathbf{e}']^T \mathbf{e}' = [\mathbf{e}']^T [\text{cov } \mathbf{d}']^{-1} \mathbf{e}' = 0.218 \quad (8.68)$$

The results may be transformed back to the original coordinates, using Equations (8.34), (8.36), (8.42), (8.45), (8.46), (8.52), and (8.56) as

$$\mathbf{G}_{MX}^{-1} = \begin{bmatrix} 0.305 & 0.167 \\ 0.173 & 0.094 \end{bmatrix}$$

$$\lambda_1 = 5.345$$

$$\begin{aligned} \mathbf{m}_{MX} &= \mathbf{S}^{-1} \mathbf{m}'_g \\ &= \begin{bmatrix} 2.054 \\ 1.163 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \hat{\mathbf{d}} &= \mathbf{D}^{-1} \hat{\mathbf{d}}' \\ &= \begin{bmatrix} 3.217 \\ 6.434 \end{bmatrix} \end{aligned}$$

$$\mathbf{e}^T \mathbf{e} = 2.670$$

$$\begin{aligned} \mathbf{e}^T [\text{cov } \mathbf{d}]^{-1} \mathbf{e} &= \begin{bmatrix} 0.783 & -1.434 \end{bmatrix} \begin{bmatrix} 0.244 & 0.032 \\ 0.032 & 0.068 \end{bmatrix} \begin{bmatrix} 0.783 \\ -1.434 \end{bmatrix} \\ &= 0.218 \end{aligned}$$

$$\hat{\mathbf{V}} = \begin{bmatrix} 0.870 & -0.707 \\ 0.493 & 0.707 \end{bmatrix}$$

$$\hat{\mathbf{U}} = \begin{bmatrix} 0.447 & -0.479 \\ 0.894 & 0.878 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.639 & -0.639 \\ 0.362 & 0.362 \end{bmatrix}$$

$$\mathbf{N} = \begin{bmatrix} 0.478 & 0.261 \\ 0.956 & 0.522 \end{bmatrix} \quad (8.69)$$

Note that $\mathbf{e}^T \mathbf{e} = 2.670$ for the weighted case is larger than the misfit $\mathbf{e}^T \mathbf{e} = 1.800$ for the unweighted case. This is to be expected because the unweighted case should produce the smallest misfit. The weighted case provides an answer that gives more weight to better known data, but it produces a larger total misfit.

The $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ matrices were obtained by transforming each eigenvector in the primed coordinate system into a vector in the original coordinates, and then scaling to unit length. Note that the vectors in $\hat{\mathbf{U}}$ (and $\hat{\mathbf{V}}$) are not perpendicular to each other. Note also that the solution \mathbf{m}_{MX} is parallel to the $[0.870, 0.493]^T$ direction in model space, also given by the first column of $\hat{\mathbf{V}}$. The predicted data $\hat{\mathbf{d}}$ is parallel to the $[0.447, 0.894]^T$ direction in data space, also given by the first column in $\hat{\mathbf{U}}$.

The resolution matrices were obtained from the primed coordinate resolution matrices after Equations (8.45)–(8.46). Note that they are no longer symmetric matrices, but that the trace has remained equal to one. The model resolution matrix \mathbf{R} still indicates that only a sum of the two model parameters m_1 and m_2 is resolved, but now we see that the estimate of m_1 is better resolved than that of m_2 . This may not seem intuitively obvious, since the *a priori* variance of m_2 is less than that of m_1 , and thus m_2 is “better known.” Because m_2 is better known, the inverse operator will leave m_2 closer to its prior estimate. Thus, m_1 will be allowed to vary further from its prior estimate. It is in this sense that the resolution of m_1 is greater than that of m_2 . The data resolution matrix \mathbf{N} still indicates that only the sum of d_1 and d_2 is resolved, or important, in constraining the solution. Now, however, the importance of the first observation has been increased significantly from the unweighted case, reflecting the smaller variance for d_1 compared to d_2 .

8.3 Damped Least Squares and the Stochastic Inverse

8.3.1 Introduction

As we have seen, the presence of small singular values causes significant stability problems with the generalized inverse. One approach is simply to set small singular values to zero, and relegate the associated eigenvectors to the zero spaces. This improves stability, with an inevitable decrease in resolution. Ideally, the cut-off value for small singular values should be based on how noisy the data are. In practice, however, the decision is almost always arbitrary.

We will now introduce a damping term, the function of which is to improve the stability of inverse problems with small singular values. First, however, we will consider another inverse operator, the *stochastic inverse*.

8.3.2 The Stochastic Inverse

Consider a forward problem given by

$$\mathbf{Gm} + \mathbf{n} = \mathbf{d} \tag{8.70}$$

where \mathbf{n} is an $N \times 1$ noise vector. It is similar to

$$\mathbf{Gm} = \mathbf{d} \tag{1.13}$$

except that we explicitly separate out the contribution of noise to the total data vector \mathbf{d} . This has some important implications, however.

We assume that both \mathbf{m} and \mathbf{n} are stochastic (i.e., random variables, as described in Chapter 2, that are characterized by their statistical properties) processes, with mean (or expected) values of zero. This is natural for noise, but implies that the mean value must be subtracted from all model parameters. Furthermore, we assume that we have estimates for the model parameter and noise covariance matrices, $[\text{cov } \mathbf{m}]$ and $[\text{cov } \mathbf{n}]$, respectively.

The stochastic inverse is defined by minimizing the average, or statistical, discrepancy between \mathbf{m} and $\mathbf{G}_s^{-1} \mathbf{d}$, where \mathbf{G}_s^{-1} is the stochastic inverse. Let $\mathbf{G}_s^{-1} = \mathbf{L}$, and determine \mathbf{L} by minimizing

$$m_i - \sum_{j=1}^N L_{ij} d_j \quad (8.71)$$

for each i . Consider repeated experiments in which \mathbf{m} and \mathbf{n} are generated. Let these values, on the k th experiment, be \mathbf{m}_k and \mathbf{n}_k , respectively. If there are a total of q experiments, then we seek \mathbf{L} which minimizes

$$\frac{1}{q} \sum_{k=1}^q \left(m_i^k - \sum_{j=1}^N L_{ij} d_j^k \right)^2 \quad (8.72)$$

The minimum of Equation (8.72) is found by differentiating with respect to L_{il} and setting it equal to zero:

$$\frac{\partial}{\partial L_{il}} \left[\frac{1}{q} \sum_{k=1}^q \left(m_i^k - \sum_{j=1}^N L_{ij} d_j^k \right)^2 \right] = 0 \quad (8.73)$$

or

$$\frac{2}{q} \sum_{k=1}^q \left(m_i^k - \sum_{j=1}^N L_{ij} d_j^k \right) (-d_l^k) = 0 \quad (8.74)$$

This implies

$$\frac{1}{q} \sum_{k=1}^q m_i^k d_l^k = \frac{1}{q} \sum_{k=1}^q \left(\sum_{j=1}^N L_{ij} d_j^k \right) d_l^k \quad (8.75)$$

The left-hand side of Equation (8.75), when taken over i and l , is simply the covariance matrix between the model parameters and the data, or

$$[\text{cov } \mathbf{m}\mathbf{d}] = \langle \mathbf{m}\mathbf{d}^T \rangle \quad (8.76)$$

The right-hand side, again taken over i and l and recognizing that \mathbf{L} will not vary from experiment to experiment, gives [see Equation (2.42c)]

$$\mathbf{L}[\text{cov } \mathbf{d}] = \mathbf{L}\langle \mathbf{d}\mathbf{d}^T \rangle \quad (8.77)$$

where $[\text{cov } \mathbf{d}]$ is the data covariance matrix. Note that $[\text{cov } \mathbf{d}]$ is not the same matrix used elsewhere in these notes. As used here, $[\text{cov } \mathbf{d}]$ is a derived quantity, based on $[\text{cov } \mathbf{m}]$ and $[\text{cov } \mathbf{n}]$. With Equations (8.76) and (8.77), we can write Equation (8.75), taken over i and l , as

$$[\text{cov } \mathbf{m}\mathbf{d}] = \mathbf{L}[\text{cov } \mathbf{d}] \quad (8.78)$$

or

$$\mathbf{L} = [\text{cov } \mathbf{m}\mathbf{d}][\text{cov } \mathbf{d}]^{-1} \quad (8.79)$$

We now need to rewrite $[\text{cov } \mathbf{d}]$ and $[\text{cov } \mathbf{m}\mathbf{d}]$ in terms of $[\text{cov } \mathbf{m}]$, $[\text{cov } \mathbf{n}]$, and \mathbf{G} . This is done as follows:

$$\begin{aligned} [\text{cov } \mathbf{d}] &= \langle \mathbf{d}\mathbf{d}^T \rangle \\ &= \langle [\mathbf{G}\mathbf{m} + \mathbf{n}][\mathbf{G}\mathbf{m} + \mathbf{n}]^T \rangle \\ &= \mathbf{G}\langle \mathbf{m}\mathbf{m}^T \rangle + \mathbf{G}\langle \mathbf{m}\mathbf{n}^T \rangle + \langle \mathbf{n}\mathbf{m}^T \rangle \mathbf{G}^T + \langle \mathbf{n}\mathbf{n}^T \rangle \end{aligned} \quad (8.80)$$

If we assume that model parameter and noise errors are uncorrelated, that is, that $\langle \mathbf{m}\mathbf{n}^T \rangle = 0 = \langle \mathbf{n}\mathbf{m}^T \rangle$, then Equation (8.80) reduces to

$$\begin{aligned} [\text{cov } \mathbf{d}] &= \mathbf{G}\langle \mathbf{m}\mathbf{m}^T \rangle \mathbf{G}^T + \langle \mathbf{n}\mathbf{n}^T \rangle \\ &= \mathbf{G}[\text{cov } \mathbf{m}]\mathbf{G}^T + [\text{cov } \mathbf{n}] \end{aligned} \quad (8.81)$$

Similarly,

$$\begin{aligned} [\text{cov } \mathbf{m}\mathbf{d}] &= \langle \mathbf{m}\mathbf{d}^T \rangle \\ &= \langle \mathbf{m}[\mathbf{G}\mathbf{m} + \mathbf{n}]^T \rangle \\ &= \langle \mathbf{m}\mathbf{m}^T \rangle \mathbf{G}^T + \langle \mathbf{m}\mathbf{n}^T \rangle \\ &= [\text{cov } \mathbf{m}]\mathbf{G}^T \end{aligned} \quad (8.82)$$

if $\langle \mathbf{m}\mathbf{n}^T \rangle = 0$.

Replacing $[\text{cov } \mathbf{m}\mathbf{d}]$ and $[\text{cov } \mathbf{d}]$ in Equation (8.79) with expressions from Equations (8.81) and (8.82), respectively, gives the definition of the *stochastic inverse operator* \mathbf{G}_s^{-1} as

$$\mathbf{G}_s^{-1} = [\text{cov } \mathbf{m}]\mathbf{G}^T \{ \mathbf{G}[\text{cov } \mathbf{m}]\mathbf{G}^T + [\text{cov } \mathbf{n}] \}^{-1} \quad (8.83)$$

Then the stochastic inverse solution, \mathbf{m}_s , is given by

$$\mathbf{m}_s = \mathbf{G}_s^{-1} \mathbf{d}$$

$$= [\text{cov } \mathbf{m}] \mathbf{G}^T [\text{cov } \mathbf{d}]^{-1} \mathbf{d} \quad (8.84)$$

It is possible to decompose the symmetric covariance matrices $[\text{cov } \mathbf{d}]$ and $[\text{cov } \mathbf{m}]$ in exactly the same manner as was done for the maximum likelihood operator [Equations (8.14) and (8.20)]:

$$[\text{cov } \mathbf{d}] = \mathbf{B} \Lambda_d \mathbf{B}^T = \{ \mathbf{B} \Lambda_d^{1/2} \} \{ \Lambda_d^{1/2} \mathbf{B}^T \} = \mathbf{D}^{-1} [\mathbf{D}^{-1}]^T \quad (8.85a)$$

$$[\text{cov } \mathbf{d}]^{-1} = \mathbf{B} \Lambda_d^{-1} \mathbf{B}^T = \mathbf{D}^T \mathbf{D} \quad (8.85b)$$

$$[\text{cov } \mathbf{m}] = \mathbf{M} \Lambda_m \mathbf{M}^T = \{ \mathbf{M} \Lambda_m^{1/2} \} \{ \Lambda_m^{1/2} \mathbf{M}^T \} = \mathbf{S}^{-1} [\mathbf{S}^{-1}]^T \quad (8.85c)$$

$$[\text{cov } \mathbf{m}]^{-1} = \mathbf{M} \Lambda_m^{-1} \mathbf{M}^T = \mathbf{S}^T \mathbf{S} \quad (8.85d)$$

where Λ_d and Λ_m are the eigenvalues of $[\text{cov } \mathbf{d}]$ and $[\text{cov } \mathbf{m}]$, respectively. The orthogonal matrices \mathbf{B} and \mathbf{M} are the associated eigenvectors.

At this point it is useful to reintroduce a set of transformations based on the decompositions in (8.85) that will transform \mathbf{d} , \mathbf{m} , and \mathbf{G} back and forth between the original coordinate system and a primed coordinate system.

$$\mathbf{m}' = \mathbf{S} \mathbf{m} \quad (8.86a)$$

$$\mathbf{d}' = \mathbf{D} \mathbf{d} \quad (8.86b)$$

$$\mathbf{G}' = \mathbf{D} \mathbf{G} \mathbf{S}^{-1} \quad (8.86c)$$

$$\mathbf{m} = \mathbf{S}^{-1} \mathbf{m}' \quad (8.87a)$$

$$\mathbf{d} = \mathbf{D}^{-1} \mathbf{d}' \quad (8.87b)$$

$$\mathbf{G} = \mathbf{D}^{-1} \mathbf{G}' \mathbf{S} \quad (8.87c)$$

Then, Equation (8.84), using primed coordinate variables, is given by

$$\begin{aligned} \mathbf{S}^{-1} \mathbf{m}'_s &= [\text{cov } \mathbf{m}] \mathbf{G}^T [\text{cov } \mathbf{d}]^{-1} \mathbf{d} \\ \mathbf{S}^{-1} \mathbf{m}'_s &= \mathbf{S}^{-1} [\mathbf{S}^{-1}]^T [\mathbf{D}^{-1} \mathbf{G}' \mathbf{S}]^T \mathbf{D}^T \mathbf{D} \mathbf{D}^{-1} \mathbf{d}' \\ &= \mathbf{S}^{-1} [\mathbf{S}^{-1}]^T \mathbf{S}^T [\mathbf{G}']^T [\mathbf{D}^{-1}]^T \mathbf{D}^T \mathbf{d}' \end{aligned} \quad (8.88a)$$

but $[\mathbf{S}^{-1}]^T \mathbf{S}^T = \mathbf{I}_M$

and $[\mathbf{D}^{-1}]^T \mathbf{D}^T = \mathbf{I}_N$

and hence $\mathbf{S}^{-1} \mathbf{m}'_s = \mathbf{S}^{-1} [\mathbf{G}']^T \mathbf{d}' \quad (8.88b)$

Premultiplying both sides by \mathbf{S} yields

$$\mathbf{m}'_s = [\mathbf{G}']^T \mathbf{d}' \quad (8.89)$$

That is, the stochastic inverse in the primed coordinate system is simply the transpose of \mathbf{G} in the primed coordinate system. Once you have found \mathbf{m}'_s , you can transform back to the original coordinates to obtain the stochastic solution as

$$\mathbf{m}_s = \mathbf{S}^{-1}\mathbf{m}'_s \quad (8.90)$$

The stochastic inverse minimizes the sum of the weighted model parameter vector and the weighted data misfit. That is, the quantity

$$\mathbf{m}^T[\text{cov } \mathbf{m}]^{-1}\mathbf{m} + [\mathbf{d} - \hat{\mathbf{d}}]^T[\text{cov } \mathbf{d}]^{-1}[\mathbf{d} - \hat{\mathbf{d}}] \quad (8.91)$$

is minimized. The generalized inverse, or maximum likelihood, minimizes both individually but not the sum.

It is important to realize that the transformations introduced in Equations (8.85), while of the same form and nomenclature as those introduced in the weighted generalized inverse case in Equations (8.14) and 8.320), differ in an important aspect. Namely, as mentioned after Equation (8.77), $[\text{cov } \mathbf{d}]$ is now a derived quantity, given by Equation (8.81):

$$[\text{cov } \mathbf{d}] = \mathbf{G}[\text{cov } \mathbf{m}]\mathbf{G}^T + [\text{cov } \mathbf{n}] \quad (8.81)$$

The data covariance matrix $[\text{cov } \mathbf{d}]$ is only equal to the noise covariance matrix $[\text{cov } \mathbf{n}]$ if you assume that the noise, or errors, in \mathbf{m} are exactly zero. Thus, before doing a stochastic inverse analysis and the transformations given in Equations (8.85), $[\text{cov } \mathbf{d}]$ must be constructed from the noise covariance matrix $[\text{cov } \mathbf{n}]$ and the mapping of model parameter uncertainties in $[\text{cov } \mathbf{m}]$ as shown in Equation (8.81).

8.3.3 Damped Least Squares

We are now ready to see how this applies to damped least squares. Suppose

$$[\text{cov } \mathbf{m}] = \sigma_m^2 \mathbf{I}_M \quad (8.92)$$

and

$$[\text{cov } \mathbf{n}] = \sigma_n^2 \mathbf{I}_N \quad (8.93)$$

Define a damping term ε^2 as

$$\varepsilon^2 = \sigma_n^2 / \sigma_m^2 \quad (8.94)$$

The stochastic inverse operator, from Equation (8.83), becomes

$$\mathbf{G}_s^{-1} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T + \varepsilon^2\mathbf{I}_N]^{-1} \quad (8.95)$$

To determine the effect of adding the ε^2 term, consider the following

$$\mathbf{G}\mathbf{G}^T = \mathbf{U}_P\Lambda_P^2\mathbf{U}_P^T \quad (7.44)$$

$[\mathbf{GG}^T]^{-1}$ exists only when $P = N$, and is given by

$$[\mathbf{GG}^T]^{-1} = \mathbf{U}_P \Lambda_P^{-2} \mathbf{U}_P^T \quad P = N \quad (7.45)$$

we can therefore write $\mathbf{GG}^T + \varepsilon^2 \mathbf{I}$ as

$$\mathbf{GG}^T + \varepsilon^2 \mathbf{I} = \begin{bmatrix} \mathbf{U}_P & | & \mathbf{U}_0 \end{bmatrix} \begin{bmatrix} \Lambda_P^2 + \varepsilon^2 \mathbf{I}_P & 0 \\ 0 & \varepsilon^2 \mathbf{I}_{N-P} \end{bmatrix} \begin{bmatrix} \mathbf{U}_P^T \\ \mathbf{U}_0^T \end{bmatrix} \quad (8.96)$$

$N \times N \quad N \times N \quad N \times N \quad N \times N$

Thus

$$[\mathbf{GG}^T + \varepsilon^2 \mathbf{I}]^{-1} = \begin{bmatrix} \mathbf{U}_P & | & \mathbf{U}_0 \end{bmatrix} \begin{bmatrix} [\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P]^{-1} & 0 \\ 0 & \varepsilon^{-2} \mathbf{I}_{N-P} \end{bmatrix} \begin{bmatrix} \mathbf{U}_P^T \\ \mathbf{U}_0^T \end{bmatrix} \quad (8.97)$$

Explicitly multiplying Equation (8.97) out gives

$$[\mathbf{GG}^T + \varepsilon^2 \mathbf{I}]^{-1} = \mathbf{U}_P [\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P]^{-1} \mathbf{U}_P^T + \mathbf{U}_0 [\varepsilon^{-2} \mathbf{I}_{N-P}] \mathbf{U}_0^T \quad (8.98)$$

Next, we write out Equation (8.95), using singular-value decomposition, as

$$\begin{aligned} \mathbf{G}_s^{-1} &= \mathbf{G}^T [\mathbf{GG}^T + \varepsilon^2 \mathbf{I}_N]^{-1} \\ &= \{ \mathbf{V}_P \Lambda_P \mathbf{U}_P^T \} \{ \mathbf{U}_P [\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P]^{-1} \mathbf{U}_P^T + \mathbf{U}_0 [\varepsilon^2 \mathbf{I}_{N-P}] \mathbf{U}_0^T \} \\ &= \mathbf{V}_P \frac{\Lambda_P}{\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P} \mathbf{U}_P^T \end{aligned} \quad (8.99)$$

since $\mathbf{U}_P^T \mathbf{U}_0 = 0$.

Note the similarity between the stochastic inverse in Equation (8.99) and the generalized inverse

$$\mathbf{G}_g^{-1} = \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \quad (7.9)$$

The net effect of the stochastic inverse is to suppress the contributions of eigenvectors with singular values less than ε . To see this, let us write out $\Lambda_P / (\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P)$ explicitly:

$$\frac{\Lambda_P}{\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P} = \begin{bmatrix} \frac{\lambda_1}{\lambda_1^2 + \varepsilon^2} & 0 & \dots & 0 \\ 0 & \frac{\lambda_2}{\lambda_2^2 + \varepsilon^2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \frac{\lambda_P}{\lambda_P^2 + \varepsilon^2} \end{bmatrix} \quad (8.100)$$

If $\lambda_i \gg \varepsilon$, then $\lambda_i / (\lambda_i^2 + \varepsilon^2) \rightarrow \lambda_i^{-1}$, the same as the generalized inverse. If $\lambda_i \ll \varepsilon$, then $\lambda_i / (\lambda_i^2 + \varepsilon) \rightarrow \lambda_i / \varepsilon^2 \rightarrow 0$. The stochastic inverse, then, dampens the contributions of eigenvectors associated with small singular values.

The stochastic inverse in Equation (8.95) looks similar to the minimum length inverse

$$\mathbf{G}_{\text{ML}}^{-1} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1} \quad (3.58)$$

To see why the stochastic inverse is also called damped least squares, consider the following:

$$\begin{aligned} [\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}_M]^{-1}\mathbf{G}^T &= \{\mathbf{V}_P[\Lambda_P^2 + \varepsilon^2\mathbf{I}_P]^{-1}\mathbf{V}_P^T + \varepsilon^2\mathbf{V}_0\mathbf{V}_0^T\}\{\mathbf{V}_P\Lambda_P\mathbf{U}_P^T\} \\ &= \{\mathbf{V}[\Lambda_P^2 + \varepsilon^2\mathbf{I}_P]^{-1}\}\{\mathbf{V}_P^T\mathbf{V}_P\Lambda_P\mathbf{U}_P^T\} + \varepsilon^{-2}\mathbf{V}_0\mathbf{V}_0^T\mathbf{V}_P\Lambda_P\mathbf{U}_P^T \\ &= \mathbf{V}_P\frac{\Lambda_P}{\Lambda_P^2 + \varepsilon^2\mathbf{I}_P}\mathbf{U}_P^T \\ &= \mathbf{G}^T[\mathbf{G}\mathbf{G}^T + \varepsilon^2\mathbf{I}_N]^{-1} \end{aligned} \quad (8.101)$$

Thus

$$[\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}_M]^{-1}\mathbf{G}^T = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T + \varepsilon^2\mathbf{I}_N]^{-1} \quad (8.102)$$

The choice of σ_m^2 is often arbitrary. Thus, ε^2 is often chosen arbitrarily to stabilize the problem. Solutions are obtained for a variety of ε^2 , and a final choice is made based on the a posteriori model covariance matrix.

The stability gained with damped least squares is not obtained without loss elsewhere. Specifically, resolution degrades with increased damping. To see this, consider the model resolution matrix for the stochastic inverse:

$$\begin{aligned} \mathbf{R} &= \mathbf{G}_s^{-1}\mathbf{G} \\ &= \mathbf{V}_P\frac{\Lambda_P^2}{\Lambda_P^2 + \varepsilon^2\mathbf{I}_P}\mathbf{V}_P^T \end{aligned} \quad (8.103)$$

It is easy to see that the stochastic inverse model resolution matrix reduces to the generalized inverse case when ε^2 goes to 0, as expected.

The reduction in model resolution can be seen by considering the trace of \mathbf{R} :

$$\text{trace}(\mathbf{R}) = \sum_{i=1}^P \frac{\lambda_i^2}{\lambda_i^2 + \varepsilon^2} \leq P \quad (8.104)$$

Similarly, the data resolution matrix \mathbf{N} is given by

$$\mathbf{N} = \mathbf{G}\mathbf{G}_s^{-1}$$

$$= \mathbf{U}_P \frac{\Lambda_P^2}{\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P} \mathbf{U}_P^T \quad (8.105)$$

$$\text{trace}(\mathbf{N}) = \sum_{i=1}^P \frac{\lambda_i^2}{\lambda_i^2 + \varepsilon^2} \leq P \quad (8.106)$$

Finally, consider the unit model covariance matrix $[\text{cov}_u \mathbf{m}]$, given by

$$\begin{aligned} [\text{cov}_u \mathbf{m}] &= \mathbf{G}_s^{-1} [\mathbf{G}_s^{-1}]^T \\ &= \mathbf{V}_P \frac{\Lambda_P^2}{[\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P]^2} \mathbf{V}_P^T \end{aligned} \quad (8.107)$$

which reduces to the generalized inverse case when $\varepsilon^2 = 0$. The introduction of ε^2 reduces the size of the covariance terms, a reflection of the stability added by including a damping term.

An alternative approach to damped least squares is achieved by adding equations of the form

$$\varepsilon m_i = 0 \quad i = 1, 2, \dots, M \quad (8.108)$$

to the original set of equations

$$\mathbf{Gm} = \mathbf{d} \quad (1.13)$$

The combined set of equations can be written in partitioned form as

$$\begin{aligned} \begin{bmatrix} \mathbf{G} \\ \varepsilon \mathbf{I}_M \end{bmatrix} \mathbf{m} &= \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \\ (N+M) \times M & \quad (N+M) \times 1 \end{aligned} \quad (8.109)$$

The least squares solution to Equation (8.109) is given by

$$\begin{aligned} \mathbf{m} &= \left\{ \begin{bmatrix} \mathbf{G}^T & \varepsilon \mathbf{I}_M \end{bmatrix} \begin{bmatrix} \mathbf{G} \\ \varepsilon \mathbf{I}_M \end{bmatrix} \right\}^{-1} \begin{bmatrix} \mathbf{G}^T & \varepsilon \mathbf{I}_M \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \\ &= [\mathbf{G}^T \mathbf{G} + \varepsilon^2 \mathbf{I}_M]^{-1} \mathbf{G}^T \mathbf{d} \end{aligned} \quad (8.110)$$

The addition of $\varepsilon^2 \mathbf{I}_M$ insures a least squares solution because $\mathbf{G}^T \mathbf{G} + \varepsilon^2 \mathbf{I}_M$ will have no eigenvalues less than ε^2 , and hence is invertible.

In signal processing, the addition of ε^2 is equivalent to adding white noise to the signal. Consider transforming

$$[\mathbf{G}^T \mathbf{G} + \varepsilon^2 \mathbf{I}_M] \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (8.111)$$

into the frequency domain as

$$[F_i^*(\omega) F_i(\omega) + \varepsilon^2]M(\omega) = F_i^*(\omega) F_o(\omega) \quad (8.112)$$

where $F_i(\omega)$ is the Fourier transform of the input waveform to some filter, * represents complex conjugate, $F_o(\omega)$ is the Fourier transform of the output wave form from the filter, $M(\omega)$ is the Fourier transform of the impulse response of the filter, and ε^2 is a constant for all frequencies ω . Solving for \mathbf{m} as the inverse Fourier transform of Equation (8.112) gives

$$\mathbf{m} = \text{F.T.}^{-1} \left[\frac{F_i^*(\omega)F_o(\omega)}{F_i^*(\omega)F_i(\omega) + \varepsilon^2} \right] \quad (8.113)$$

The addition of ε^2 in the denominator assures that the solution is not dominated by small values of $F_i(\omega)$, which can arise when the signal-to-noise ratio is poor. Because the ε^2 term is added equally at all frequencies, this is equivalent to adding white light to the signal.

Damping is particularly useful in nonlinear problems. In nonlinear problems, small singular values can produce very large changes, or steps, during the iterative process. These large steps can easily violate the assumption of linearity in the region where the nonlinear problem was linearized. In order to limit step sizes, an ε^2 term can be added. Typically, one uses a fairly large value of ε^2 during the initial phase of the iterative procedure, gradually letting ε^2 go to zero as the solution is approached.

Recall that the generalized inverse minimized $[\mathbf{d} - \mathbf{G}\mathbf{m}]^T[\mathbf{d} - \mathbf{G}\mathbf{m}]$ and $\mathbf{m}^T\mathbf{m}$ individually. Consider a new function E to minimize, defined by

$$\begin{aligned} E &= [\mathbf{d} - \mathbf{G}\mathbf{m}]^T[\mathbf{d} - \mathbf{G}\mathbf{m}] + \varepsilon^2\mathbf{m}^T\mathbf{m} \\ &= \mathbf{m}^T\mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{m}^T\mathbf{G}^T\mathbf{d} - \mathbf{d}^T\mathbf{G}\mathbf{m} + \mathbf{d}^T\mathbf{d} + \varepsilon^2\mathbf{m}^T\mathbf{m} \end{aligned} \quad (8.114)$$

Differentiating E with respect to \mathbf{m}^T and setting it equal to zero yields

$$\partial E / \partial \mathbf{m}^T = \mathbf{G}^T\mathbf{G}\mathbf{m} - \mathbf{G}^T\mathbf{d} + \varepsilon^2\mathbf{m} = 0$$

or

$$[\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}_M]\mathbf{m} = \mathbf{G}^T\mathbf{d} \quad (8.115)$$

This shows why damped least squares minimized a weighted sum of the misfit and the length of the model parameter vector.

8.4 Ridge Regression

8.4.1 Mathematical Background

Recall the least squares operator

$$[\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T \quad (8.116)$$

If the data covariance matrix $[\text{cov } \mathbf{d}]$ is given by

$$[\text{cov } \mathbf{d}] = \sigma^2\mathbf{I} \quad (8.117)$$

then the a posteriori model covariance matrix $[\text{cov } \mathbf{m}]$, also called the dispersion of \mathbf{m} , is given by

$$[\text{cov } \mathbf{m}] = \sigma^2[\mathbf{G}^T\mathbf{G}]^{-1} \quad (8.118)$$

In terms of singular-value decomposition, it is given by

$$[\text{cov } \mathbf{m}] = \sigma^2\mathbf{V}_p\mathbf{\Lambda}_p^{-2}\mathbf{V}_p^T. \quad (8.119)$$

This can also be written as

$$[\text{cov } \mathbf{m}] = \sigma^2\left[\mathbf{V}_p|\mathbf{V}_0\right]\left[\begin{array}{c|c} \mathbf{\Lambda}_p^{-2} & \\ \hline & \mathbf{0} \end{array}\right]\left[\begin{array}{c} \mathbf{V}_p^T \\ \mathbf{V}_0 \end{array}\right] \quad (8.120)$$

The total variance is defined as the trace of the model covariance matrix, given by

$$\text{trace} [\text{cov } \mathbf{m}] = \sigma^2 \{\text{trace} [\mathbf{G}^T\mathbf{G}]^{-1}\} = \sigma^2 \sum_{i=1}^P \frac{1}{\lambda_i^2} \quad (8.121)$$

which follows from the fact that the trace of a matrix is invariant under an orthogonal coordinate transformation.

It is clear from Equation (8.121) that the total variance will get large as λ_i gets small. We saw that the stochastic inverse operator

$$\mathbf{G}_s^{-1} = [\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}_M]^{-1}\mathbf{G}^T = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T + \varepsilon^2\mathbf{I}_N]^{-1} \quad (8.102)$$

resulted in a reduction of the model covariance (8.107). In fact, the addition of ε^2 to each diagonal entry $\mathbf{G}^T\mathbf{G}$ results in a total variance defined by

$$\text{trace} [\text{cov } \mathbf{m}] = \sigma^2\{\text{trace} [\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}]^{-1}\} = \sigma^2 \sum_{i=1}^P \frac{\lambda_i^2}{(\lambda_i^2 + \varepsilon^2)^2} \quad (8.122)$$

Clearly, Equation (8.122) is less than (8.121) for all $\varepsilon^2 > 0$.

8.4.2 The Ridge Regression Operator

The stochastic inverse operator of Equation (8.102) is also called ridge regression for reasons that I will explain shortly. The ridge regression operator is derived as follows. We seek an

operator that finds a solution \mathbf{m}_{RR} that is closest to the origin (as in the minimum length case), subject to the constraint that the solution lie on an ellipsoid defined by

$$\begin{matrix} [\mathbf{m}_{RR} - \mathbf{m}_{LS}]^T & \mathbf{G}^T \mathbf{G} & [\mathbf{m}_{RR} - \mathbf{m}_{LS}] & = & \phi_0 \\ 1 \times M & M \times M & M \times 1 & & 1 \times 1 \end{matrix} \quad (8.123)$$

where \mathbf{m}_{LS} is the least squares solution (i.e., obtained by setting ε^2 equal to 0). Equation (8.123) represents a single-equation quadratic in \mathbf{m}_{RR} .

The ridge regression operator \mathbf{G}_{RR}^{-1} is obtained using Lagrange multipliers. We form the function

$$\Psi(\mathbf{m})_{RR} = \mathbf{m}_{RR}^T \mathbf{m}_{RR} + \lambda \{ [\mathbf{m}_{RR} - \mathbf{m}_{LS}]^T \mathbf{G}^T \mathbf{G} [\mathbf{m}_{RR} - \mathbf{m}_{LS}] - \phi_0 \} \quad (8.124)$$

and differentiate with respect to \mathbf{m}_{RR}^T to obtain

$$\mathbf{m}_{RR} + \lambda \mathbf{G}^T \mathbf{G} [\mathbf{m}_{RR} - \mathbf{m}_{LS}] = 0 \quad (8.125)$$

Solving Equation (8.125) for \mathbf{m}_{RR} gives

$$[\lambda \mathbf{G}^T \mathbf{G} + \mathbf{I}_M] \mathbf{m}_{RR} = \lambda \mathbf{G}^T \mathbf{G} \mathbf{m}_{LS}$$

or

$$\mathbf{m}_{RR} = [\lambda \mathbf{G}^T \mathbf{G} + \mathbf{I}_M]^{-1} \lambda \mathbf{G}^T \mathbf{G} \mathbf{m}_{LS} \quad (8.126)$$

The least squares solution \mathbf{m}_{LS} is given by

$$\mathbf{m}_{LS} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \quad (3.27)$$

Substituting \mathbf{m}_{LS} from Equation (3.27) into (8.126)

$$\begin{aligned} \mathbf{m}_{RR} &= [\lambda \mathbf{G}^T \mathbf{G} + \mathbf{I}_M]^{-1} \lambda \mathbf{G}^T \mathbf{G} [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T \mathbf{d} \\ &= [\lambda \mathbf{G}^T \mathbf{G} + \mathbf{I}_M]^{-1} \lambda \mathbf{G}^T \mathbf{d} \\ &= \frac{1}{\lambda} \left[\mathbf{G}^T \mathbf{G} + \frac{1}{\lambda} \mathbf{I}_M \right]^{-1} \lambda \mathbf{G}^T \mathbf{d} \quad \lambda \neq 0 \\ &= \left[\mathbf{G}^T \mathbf{G} + \frac{1}{\lambda} \mathbf{I}_M \right]^{-1} \mathbf{G}^T \mathbf{d} \end{aligned} \quad (8.127)$$

If we let $1/\lambda = \varepsilon^2$, then Equation (8.127) becomes

$$\mathbf{m}_{RR} = [\mathbf{G}^T \mathbf{G} + \varepsilon^2 \mathbf{I}_M]^{-1} \mathbf{G}^T \mathbf{d} \quad (8.128)$$

and the ridge regression operator \mathbf{G}_{RR}^{-1} is defined as

$$\mathbf{G}_{RR}^{-1} = [\mathbf{G}^T \mathbf{G} + \varepsilon^2 \mathbf{I}_M]^{-1} \mathbf{G}^T \quad (8.129)$$

In terms of singular-value decomposition, the ridge regression operator \mathbf{G}_{RR}^{-1} is identical to the stochastic inverse operator, and following Equation (8.99),

$$\mathbf{G}_{RR}^{-1} = \mathbf{V}_P \frac{\Lambda_P}{\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P} \mathbf{U}_P^T \quad (8.130)$$

In practice, we determine ε^2 (and thus λ) by trial and error, with the attendant trade-off between resolution and stability. As defined, however, every choice of ε^2 is associated with a particular ϕ_0 and hence a particular ellipsoid from Equation (8.123). Changing ϕ_0 does not change the orientation of the ellipsoid; it simply stretches or contracts the major and minor axes. We can think of the family of ellipsoids defined by varying ε^2 (or ϕ_0) as a ridge in solution space, with each particular ε^2 (or ϕ_0) being a contour of the ridge. We then obtain the the ridge regression solution by following one of the contours around the ellipsoid until we find the point closest to the origin, hence the name ridge regression.

8.4.3 An Example of Ridge Regression Analysis

A simple example will help clarify the ridge regression operator. Consider the following:

$$\begin{array}{ccc} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} & = & \begin{bmatrix} 8 \\ 4 \end{bmatrix} \\ \mathbf{G} & \mathbf{m} & & \mathbf{d} \end{array} \quad (8.131)$$

Singular-value decomposition gives

$$\begin{array}{l} \mathbf{U}_P = \mathbf{U} = \mathbf{I}_2 \\ \mathbf{V}_P = \mathbf{V} = \mathbf{I}_2 \\ \Lambda_P = \Lambda = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \quad (8.132)$$

The generalized inverse \mathbf{G}_g^{-1} is given by

$$\begin{aligned} \mathbf{G}_g^{-1} &= \mathbf{V}_P \Lambda_P^{-1} \mathbf{U}_P^T \\ &= \mathbf{I}_2 \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{I}_2^T \\ &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (8.133)$$

The generalized inverse solution (also the exact, or least squares, solution) is

$$\mathbf{m}_{LS} = \mathbf{G}_g^{-1} \mathbf{d} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \quad (8.134)$$

The ridge regression solution is given by

$$\begin{aligned} \mathbf{m}_{RR} &= \mathbf{G}_{RR}^{-1} \mathbf{d} = \mathbf{V}_P \frac{\Lambda_P}{\Lambda_P^2 + \varepsilon^2 \mathbf{I}_P} \mathbf{U}_P^T \begin{bmatrix} 8 \\ 4 \end{bmatrix} \\ &= \mathbf{I}_2 \begin{bmatrix} \frac{2}{4 + \varepsilon^2} & 0 \\ 0 & \frac{1}{1 + \varepsilon^2} \end{bmatrix} \mathbf{I}_2 \begin{bmatrix} 8 \\ 4 \end{bmatrix} \\ &= \begin{bmatrix} \frac{2}{4 + \varepsilon^2} & 0 \\ 0 & \frac{1}{1 + \varepsilon^2} \end{bmatrix} \begin{bmatrix} 8 \\ 4 \end{bmatrix} \end{aligned} \quad (8.135)$$

Note that for $\varepsilon^2 = 0$, the least squares solution is recovered. Also, as $\varepsilon^2 \rightarrow \infty$, the solution goes to the origin. Thus, as expected, the solution varies from the least squares solution to the origin as more and more weight is given to minimizing the length of the solution vector.

We can now determine the ellipsoid associated with a particular value of ε^2 . For example, let $\varepsilon^2 = 1$. Then the ridge regression solution, from Equation (8.135), is

$$\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}_{RR} = \begin{bmatrix} \frac{16}{4 + \varepsilon^2} \\ \frac{4}{1 + \varepsilon^2} \end{bmatrix} = \begin{bmatrix} 3.2 \\ 2 \end{bmatrix} \quad (8.136)$$

Now, returning to the constraint Equation (8.123), we have that

$$\begin{bmatrix} m_1 - 4.0 \\ m_2 - 4.0 \end{bmatrix}^T \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 - 4.0 \\ m_2 - 4.0 \end{bmatrix} = \phi_0$$

or

$$4(m_1 - 4.0)^2 + (m_2 - 4.0)^2 = \phi_0 \quad (8.137)$$

To find ϕ_0 , we substitute the solution from Equation (8.136) into (8.123) and

$$4(3.2 - 4.0)^2 + (2.0 - 4.0)^2 = \phi_0$$

or

$$\phi_0 = 6.56 \quad (8.138)$$

Substituting ϕ_0 from Equation (8.138) back into (8.137) and rearranging gives

$$\frac{(m_1 - 4.0)^2}{1.64} + \frac{(m_2 - 4.0)^2}{6.56} = 1.0 \quad (8.139)$$

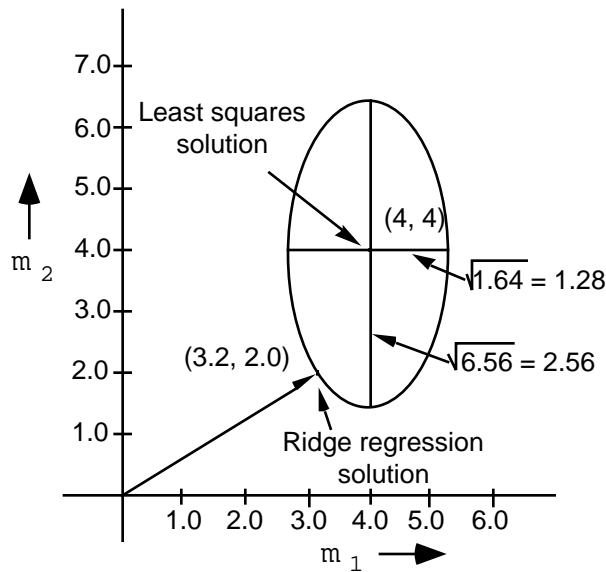
Equation (8.139) is of the form

$$\frac{(x - h)^2}{b^2} + \frac{(y - k)^2}{a^2} = 1.0 \quad (8.140)$$

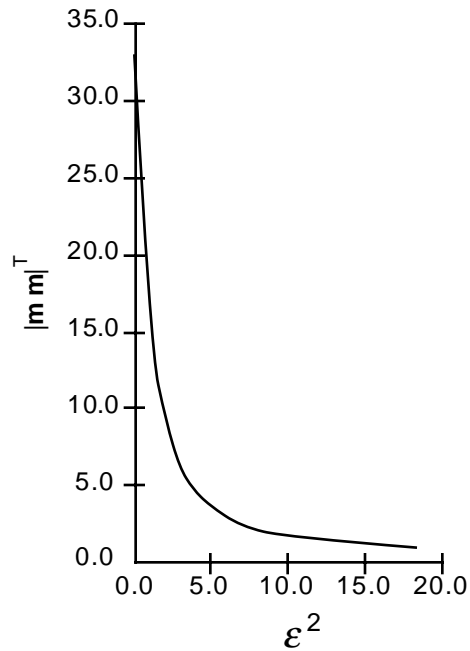
which represents an ellipse centered at (h, k) , with semimajor and semiminor axes a and b parallel to the y and x axes, respectively. Thus, for the current example, the lengths of the semimajor and semiminor axes are 2.56 and 1.28, respectively. The axes of the ellipse are parallel to the m_2 and m_1 axes, and the ellipse is centered at $(4, 4)$. Different choices for ϵ^2 will produce a family of ellipses centered on $(4, 4)$, with semimajor and semiminor axes parallel to the m_2 and m_1 axes, respectively, and with the semimajor axis always twice the length of the semiminor axis.

The shape and orientation of the family of ellipses follow completely from the structure of the original \mathbf{G} matrix. The axes of the ellipse coincide with the m_1 and m_2 axes because the original \mathbf{G} matrix was diagonal. If the original \mathbf{G} matrix had not been diagonal, the axes of the ellipse would have been inclined to the m_1 and m_2 axes. The center of the ellipse, given by the least squares solution, is, of course, both a function of \mathbf{G} and the data vector \mathbf{d} .

The graph below illustrates this particular problem for $\epsilon^2 = 1$.

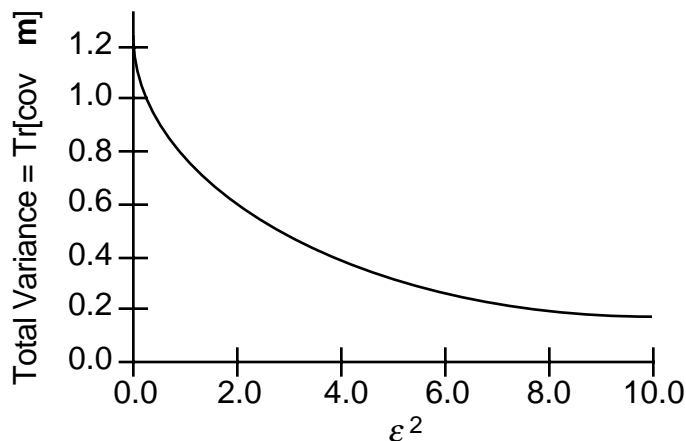


It is also instructive to plot the length squared of the solution, $\mathbf{m}^T \mathbf{m}$, as a function of ϵ^2 :



This figure shows that adding ϵ^2 damps the solution from least squares toward zero length as ϵ^2 increases.

Next consider a plot of the total variance from Equation (8.122) as a function of ϵ^2 for data variance $\sigma^2 = 1$.

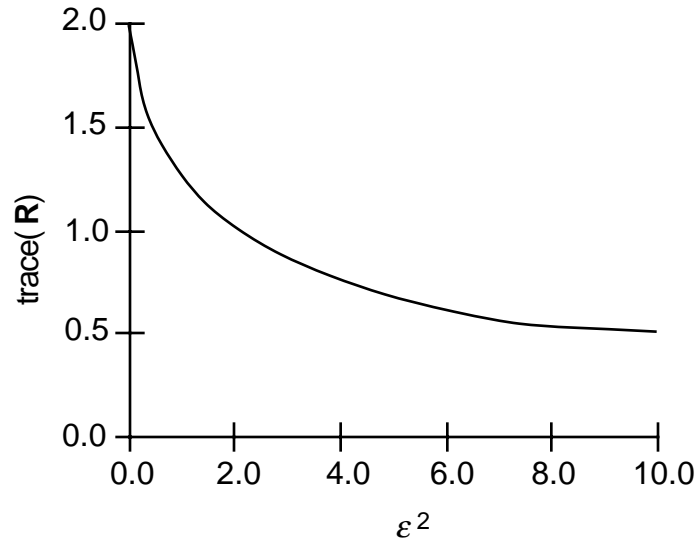


The total variance decreases, as expected, as more damping is included.

Finally, consider the model resolution matrix \mathbf{R} given by

$$\begin{aligned} \mathbf{R} &= \mathbf{G}_{RR}^{-1} \mathbf{G} \\ &= \mathbf{V}_P \frac{\Lambda_P^2}{\Lambda_P^2 + \epsilon^2 \mathbf{I}_P} \mathbf{V}_P^T \end{aligned} \tag{8.141}$$

We can plot trace (\mathbf{R}) as a function of ε^2 and get



For $\varepsilon^2 = 0$, we have perfect model resolution, with $\text{trace}(\mathbf{R}) = P = 2 = M = N$. As ε^2 increases, the model resolution decreases. Comparing the plots of total variance and the trace of the model resolution matrix, we see that as ε^2 increases, stability improves (total variance decreases) while resolution degrades. This is an inevitable trade-off.

In this particular simple example, it is hard to choose the most appropriate value for ε^2 because, in fact, the sizes of the two singular values differ very little. In general, when the singular values differ greatly, the plots for total variance and trace (\mathbf{R}) can help us choose ε^2 . If the total variance initially diminishes rapidly and then very slowly for increasing ε^2 , choosing ε^2 near the bend in the total variance curve is most appropriate.

We have shown in this section how the ridge regression operator is formed and how it is equivalent to damped least squares and the stochastic inverse operator.

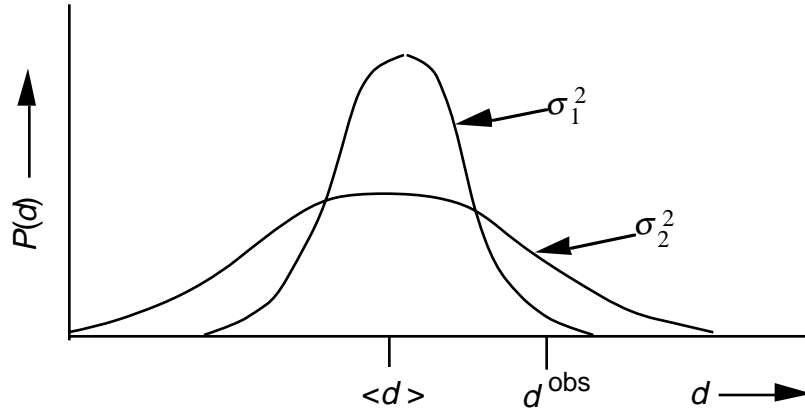
8.5 Maximum Likelihood

8.5.1 Background

The maximum likelihood approach is fundamentally probabilistic in nature. A probability density function (PDF) is created in data space that assigns a probability $P(\mathbf{d})$ to every point in data space. This PDF is a function of the model parameters, and hence $P(\mathbf{d})$ may change with each choice of \mathbf{m} . The underlying principle of the maximum likelihood approach is to find a solution \mathbf{m}_{MX} such that $P(\mathbf{d})$ is maximized at the observed data \mathbf{d}^{obs} . Put another way, a solution \mathbf{m}_{MX} is sought such that the probability of observing the observed data is maximized. At first thought, this may not seem very satisfying. After all, in some sense there is a 100% chance that the observed data are observed, simply because they are the observed data. The point is, however, that $P(\mathbf{d})$ is a calculated quantity, which varies over data space as a function of \mathbf{m} . Put this way, does it make sense to choose \mathbf{m} such that $P(\mathbf{d}^{\text{obs}})$ is small, meaning that the observed data are an unlikely

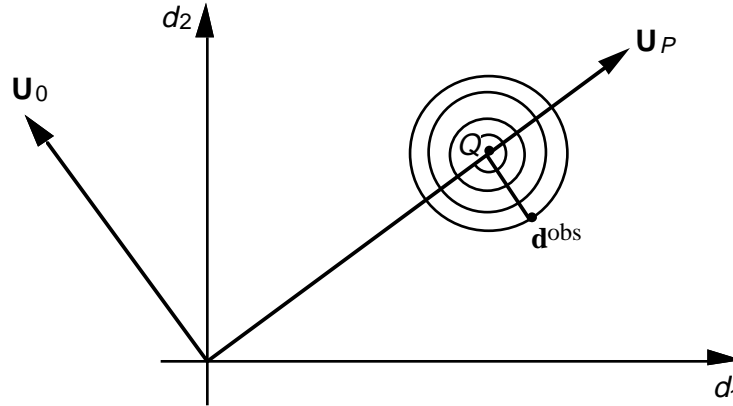
outcome of some experiment? This is clearly not ideal. Rather, it makes more sense to choose \mathbf{m} such that $P(\mathbf{d}^{\text{obs}})$ is as large as possible, meaning that you have found an \mathbf{m} for which the observed data, which exist with 100% certainty, are as likely an outcome as possible.

Imagine a very simple example with a single observation where $P(d)$ is Gaussian with fixed mean $\langle d \rangle$ and a variance σ^2 that is a function of some model parameter m . For the moment we need not worry about how m affects σ^2 , other than to realize that as m changes, so does σ^2 . Consider the diagram below, where the vertical axis is probability, and the horizontal axis is d . Shown on the diagram are d^{obs} , the observed datum; $\langle d \rangle$, the mean value for the Gaussian $P(d)$; and two different $P(d)$ curves based on two different variance estimates σ_1^2 and σ_2^2 , respectively.



The area under both $P(d)$ curves is equal to one, since this represents integrating $P(d)$ over all possible data values. The curve for σ_1^2 , where σ_1^2 is small, is sharply peaked at $\langle d \rangle$, but is very small at d^{obs} . In fact, d^{obs} appears to be several standard deviations σ from $\langle d \rangle$, indicating that d^{obs} is a very unlikely outcome. $P(d)$ for σ_2^2 , on the other hand, is not as sharply peaked at $\langle d \rangle$, but because the variance is larger, $P(d)$ is larger at the observed datum, d^{obs} . You could imagine letting σ^2 get very large, in which case values far from $\langle d \rangle$ would have $P(d)$ larger than zero, but no value of $P(d)$ would be very large. In fact, you could imagine $P(d^{\text{obs}})$ becoming smaller than the case for σ_2^2 . Thus, the object would be to vary m , and hence σ^2 , such that $P(d^{\text{obs}})$ is maximized. Of course, for this simple example we have not worried about the mechanics of finding m , but we will later for more realistic cases.

A second example, after one near the beginning of Menke's Chapter 5, is also very illustrative. Imagine collecting a single datum N times in the presence of Gaussian noise. The observed data vector \mathbf{d}^{obs} has N entries and hence lies in an N -dimensional data space. You can think of each observation as a random variable with the same mean $\langle d \rangle$ and variance σ^2 , both of which are unknown. The goal is to find $\langle \mathbf{d} \rangle$ and σ^2 . We can cast this problem in our familiar $\mathbf{Gm} = \mathbf{d}$ form by associating \mathbf{m} with $\langle \mathbf{d} \rangle$ and noting that $\mathbf{G} = (1/N)[1, 1, \dots, 1]^T$. Consider the simple case where $N = 2$, shown on the next page:



The observed data \mathbf{d}^{obs} are a point in the d_1d_2 plane. If we do singular-value decomposition on \mathbf{G} , we see immediately that, in general, $\mathbf{U}_P = (1/\sqrt{N})[1, 1, \dots, 1]^T$, and for our $N=2$ case, $\mathbf{U}_P = [1/\sqrt{2}, 1/\sqrt{2}]^T$, and $\mathbf{U}_0 = [-1/\sqrt{2}, 1/\sqrt{2}]^T$. We recognize that all predicted data must lie in \mathbf{U}_P space, which is a single vector. Every choice of $\mathbf{m} = \langle d \rangle$ gives a point on the line $d_1 = d_2 = \dots = d_N$. If we slide $\langle d \rangle$ up to the point Q on the diagram, we see that all the misfit lies in \mathbf{U}_0 space, and we have obtained the least squares solution for $\langle d \rangle$. Also shown on the figure are contours of $P(\mathbf{d})$ based on σ^2 . If σ^2 is small, the contours will be close together, and $P(\mathbf{d}^{\text{obs}})$ will be small. The contours are circular because the variance is the same for each d_i . Our $N=2$ case has thus reduced to the one-dimensional case discussed on the previous page, where some value of σ^2 will maximize $P(\mathbf{d}^{\text{obs}})$. Menke (Chapter 5) shows that $P(\mathbf{d})$ for the N -dimensional case with Gaussian noise is given by

$$P(d) = \frac{1}{\sigma^N (2\pi)^{N/2}} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^N (d_i - \langle d \rangle)^2 \right] \quad (8.142)$$

where d_i are the observed data and $\langle d \rangle$ and σ are the unknown model parameters. The solution for $\langle d \rangle$ and σ is obtained by maximizing $P(\mathbf{d})$. That is, the partials of $P(\mathbf{d})$ with respect to $\langle d \rangle$ and σ are formed and set to zero. Menke shows that this leads to

$$\langle d \rangle^{\text{est}} = \frac{1}{N} \sum_{i=1}^N d_i \quad (8.143)$$

$$\sigma^{\text{est}} = \left[\frac{1}{N} \sum_{i=1}^N (d_i - \langle d \rangle)^2 \right]^{1/2} \quad (8.144)$$

We see that $\langle d \rangle$ is found independently of σ , and this shows why the least squares solution (point Q on the diagram) seems to be found independently of σ . Now, however, Equation (8.144) indicates that σ^{est} will vary for different choices of $\langle d \rangle$ affecting $P(\mathbf{d}^{\text{obs}})$.

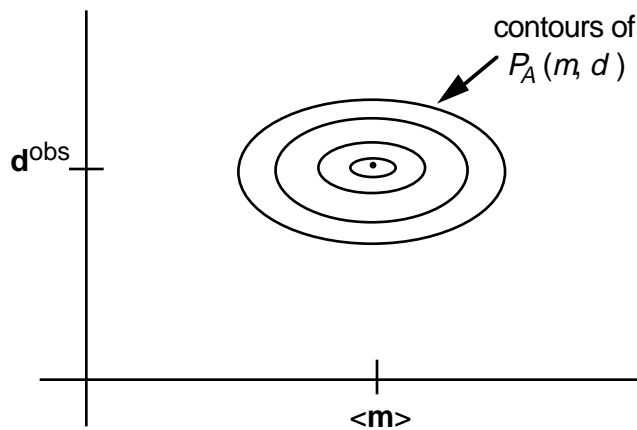
The example can be extended to the general data vector \mathbf{d} case where the Gaussian noise (possibly correlated) in \mathbf{d} is described by the data covariance matrix $[\text{cov } \mathbf{d}]$. Then it is possible to assume that $P(\mathbf{d})$ has the form

$$P(\mathbf{d}) \propto \exp \left\{ -\frac{1}{2} [\mathbf{d} - \mathbf{G}\mathbf{m}]^T [\text{cov } \mathbf{d}]^{-1} [\mathbf{d} - \mathbf{G}\mathbf{m}] \right\} \quad (8.145)$$

We note that the exponential in Equation (8.145) reduces to the exponential in Equation (8.142) when $[\text{cov } \mathbf{d}] = \sigma^2 \mathbf{I}$, and $\mathbf{G}\mathbf{m}$ gives the predicted data, given by $\langle d \rangle$. $P(\mathbf{d})$ in Equation (8.145) is maximized when $[\mathbf{d} - \mathbf{G}\mathbf{m}]^T [\text{cov } \mathbf{d}]^{-1} [\mathbf{d} - \mathbf{G}\mathbf{m}]$ is minimized. This is, of course, exactly what is minimized in the weighted least squares [Equations (3.59) and (3.60)] and weighted generalized inverse [Equation (8.10)] approaches. We can make the very important conclusion that maximum likelihood approaches are equivalent to weighted least squares or weighted generalized inverse approaches when the noise in the data is Gaussian.

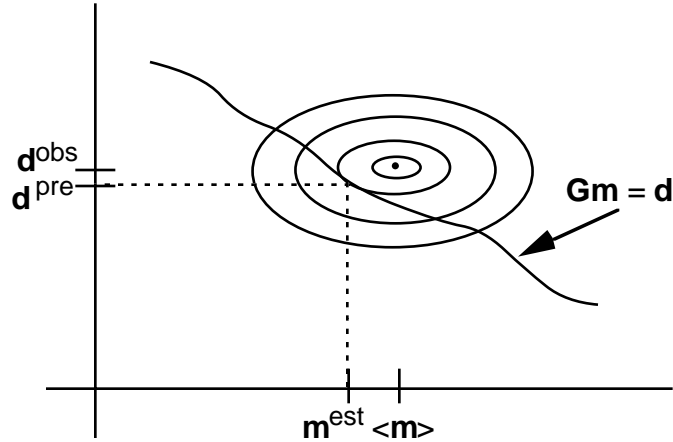
8.5.2 The General Case

We found in the generalized inverse approach that whenever $P < M$, the solution is nonunique. The equivalent viewpoint with the maximum likelihood approach is that $P(\mathbf{d})$ does not have a well-defined peak. In this case, prior information (such as minimum length for the generalized inverse) must be added. We can think of \mathbf{d}^{obs} and $[\text{cov } \mathbf{d}]$ as prior information for the data, which we could summarize as $P_A(\mathbf{d})$. The prior information about the model parameters could also be summarized as $P_A(\mathbf{m})$ and could take the form of a prior estimate of the solution $\langle m \rangle$ and a covariance matrix $[\text{cov } \mathbf{m}]_A$. Graphically (after Figure 5.9 in Menke) you can represent the joint distribution $P_A(m, d) = P_A(m) P_A(d)$ detailing the prior knowledge of data and model spaces as



where $P_A(m, d)$ is contoured about $(\mathbf{d}^{\text{obs}}, \langle m \rangle)$, the most likely point in the prior distribution. The contours are not inclined to the model or data axes because we assume that there is no correlation between our prior knowledge of \mathbf{d} and \mathbf{m} . As shown, the figure indicates less confidence in $\langle m \rangle$ than in the data. Of course, if the maximum likelihood approach were applied to $P_A(m, d)$, it would return $(\mathbf{d}^{\text{obs}}, \langle m \rangle)$ because there has not been any attempt to include the forward problem $\mathbf{G}\mathbf{m} = \mathbf{d}$.

Each choice of \mathbf{m} leads to a predicted data vector \mathbf{d}^{pre} . In the schematic figure on the next page, the forward problem $\mathbf{G}\mathbf{m} = \mathbf{d}$ is thus shown as a line in the model space–data space plane:



The maximum likelihood solution \mathbf{m}^{est} is the point where the $P(\mathbf{d})$ obtains its maximum value along the $\mathbf{G}\mathbf{m} = \mathbf{d}$ curve. If you imagine that $P(\mathbf{d})$ is very elongated along the model-space axis, this is equivalent to saying that the data are known much better than the prior model parameter estimate $\langle \mathbf{m} \rangle$. In this case \mathbf{d}^{pre} will be very close to the observed data \mathbf{d}^{obs} , but the estimated solution \mathbf{m}^{est} may be very far from $\langle \mathbf{m} \rangle$. Conversely, if $P(\mathbf{d})$ is elongated along the data axis, then the data uncertainties are relatively large compared to the confidence in $\langle \mathbf{m} \rangle$, and \mathbf{m}^{est} will be close to $\langle \mathbf{m} \rangle$, while \mathbf{d}^{pre} may be quite different from \mathbf{d}^{obs} .

Menke also points out that there may be uncertainties in the theoretical forward relationship $\mathbf{G}\mathbf{m} = \mathbf{d}$. These may be expressed in terms of an $N \times N$ inexact-theory covariance matrix $[\text{cov } \mathbf{g}]$. This covariance matrix deserves some comment. As in any covariance matrix of a single term (e.g., \mathbf{d} , \mathbf{m} , or \mathbf{G}), the diagonal entries are variances, and the off-diagonal terms are covariances. What does the (1, 1) entry of $[\text{cov } \mathbf{g}]$ refer to, however? It turns out to be the variance of the first equation (row) in \mathbf{G} . Similarly, each diagonal term in $[\text{cov } \mathbf{g}]$ refers to an uncertainty of a particular equation (row) in \mathbf{G} , and off-diagonal terms are covariances between rows in \mathbf{G} . Each row in \mathbf{G} times \mathbf{m} gives a predicted datum. For example, the first row of \mathbf{G} times \mathbf{m} gives d_1^{pre} . Thus a large variance for the (1, 1) term in $[\text{cov } \mathbf{g}]$ would imply that we do not have much confidence in the theory's ability to predict the first observation. It is easy to see that this is equivalent to saying that not much weight should be given to the first observation. We will see, then, that $[\text{cov } \mathbf{g}]$ plays a role similar to $[\text{cov } \mathbf{d}]$.

We are now in a position to give the maximum likelihood operator $\mathbf{G}_{\text{MX}}^{-1}$ in terms of \mathbf{G} , and the data ($[\text{cov } \mathbf{d}]$), model parameter ($[\text{cov } \mathbf{m}]$), and theory ($[\text{cov } \mathbf{g}]$) covariance matrices as

$$\mathbf{G}_{\text{MX}}^{-1} = [\text{cov } \mathbf{m}]^{-1} \mathbf{G}^T \{ [\text{cov } \mathbf{d}] + [\text{cov } \mathbf{g}] + \mathbf{G} [\text{cov } \mathbf{m}]^{-1} \mathbf{G}^T \}^{-1} \quad (8.146a)$$

$$= [\mathbf{G}^T \{ [\text{cov } \mathbf{d}] + [\text{cov } \mathbf{g}] \}^{-1} \mathbf{G} + [\text{cov } \mathbf{m}]^{-1}]^{-1} \mathbf{G}^T \{ [\text{cov } \mathbf{d}] + [\text{cov } \mathbf{g}] \}^{-1} \quad (8.146b)$$

where Equations (8.146a) and (8.146b) are equivalent. There are several points to make. First, as mentioned previously, $[\text{cov } \mathbf{d}]$ and $[\text{cov } \mathbf{g}]$ appear everywhere as a pair. Thus, the two covariance matrices play equivalent roles. Second, if we ignore all of the covariance information, we see that Equation (8.146a) looks like $\mathbf{G}^T [\mathbf{G}\mathbf{G}^T]^{-1}$, which is the minimum length operator. Third, if we again ignore all covariance information, Equation (8.146b) looks like $[\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$, which is the least squares operator. Thus, we see that the maximum likelihood operator can be viewed as some kind of a combined weighted least squares and weighted minimum length operator.

The maximum likelihood solution \mathbf{m}_{MX} is given by

$$\mathbf{m}_{MX} = \langle \mathbf{m} \rangle + \mathbf{G}_{MX}^{-1} [\mathbf{d} - \mathbf{G} \langle \mathbf{m} \rangle] \quad (8.147)$$

$$\begin{aligned} &= \langle \mathbf{m} \rangle + \mathbf{G}_{MX}^{-1} \mathbf{d} - \mathbf{G}_{MX}^{-1} \mathbf{G} \langle \mathbf{m} \rangle \\ &= \mathbf{G}_{MX}^{-1} \mathbf{d} + [\mathbf{I} - \mathbf{R}] \langle \mathbf{m} \rangle \end{aligned} \quad (8.148)$$

where \mathbf{R} is the model resolution matrix. Equation (8.148) explicitly shows the dependence of \mathbf{m}_{MX} on the prior estimate of the solution $\langle \mathbf{m} \rangle$. If there is perfect model resolution, then $\mathbf{R} = \mathbf{I}$, and \mathbf{m}_{MX} is independent of $\langle \mathbf{m} \rangle$. If the i th row of \mathbf{R} is equal to the i th row of the identity matrix, then there will be no dependence on the i th entry in \mathbf{m}_{MX} on the i th entry in $\langle \mathbf{m} \rangle$.

Menke points out that there are several interesting limiting cases for the maximum likelihood operator. We begin by assuming some simple forms for the covariance matrices:

$$[\text{cov } \mathbf{g}] = \sigma_g^2 \mathbf{I}_N \quad (8.149a)$$

$$[\text{cov } \mathbf{m}] = \sigma_m^2 \mathbf{I}_M \quad (8.149b)$$

$$[\text{cov } \mathbf{d}] = \sigma_d^2 \mathbf{I}_N \quad (8.149c)$$

In the first case we assume that the data and theory are much better known than $\langle \mathbf{m} \rangle$. In the limiting case we can assume $\sigma_d^2 = \sigma_g^2 = 0$. If we do, then Equation (8.146a) reduces to $\mathbf{G}_{MX}^{-1} = \mathbf{G}^T [\mathbf{G} \mathbf{G}^T]^{-1}$, the minimum length operator. If we assume that $[\text{cov } \mathbf{m}]$ still has some structure, then Equation (8.146a) reduces to

$$\mathbf{G}_{MX}^{-1} = [\text{cov } \mathbf{m}]^{-1} \mathbf{G}^T \{ \mathbf{G} [\text{cov } \mathbf{m}]^{-1} \mathbf{G}^T \} \quad (8.150)$$

the weighted minimum length operator. If we assume only that σ_d^2 and σ_g^2 are much less than σ_m^2 and that $1/\sigma_m^2$ goes to 0, then Equation (8.146b) reduces to $\mathbf{G}_{MX}^{-1} = [\mathbf{G}^T \mathbf{G}]^{-1} \mathbf{G}^T$, or the least squares operator. It is important to realize that $[\mathbf{G}^T \mathbf{G}]^{-1}$ only exists when $P = M$, and $[\mathbf{G} \mathbf{G}^T]^{-1}$ only exists when $P = N$. Thus, either form, or both, may fail to exist, depending on P . The simplifying assumptions about σ_d^2 , σ_g^2 , and σ_m^2 can thus break down the equivalence between Equations (8.146a) and (8.146b).

A second limiting case involves assuming no confidence in either (or both) the data or theory. That is, we let σ_d^2 and/or σ_g^2 go to infinity. Then we see that \mathbf{G}_{MX}^{-1} goes to $\mathbf{0}$ and $\mathbf{m}_{MX} = \langle \mathbf{m} \rangle$. This makes sense if we realize that we have assumed the data are useless (and/or the theory), and hence we do not have a useful forward problem to move us away from our prior estimate $\langle \mathbf{m} \rangle$.

We have assumed in deriving Equations (8.146a) and (8.146b) that all of the covariance matrices represent Gaussian processes. In this case, we have shown that maximum likelihood approaches will yield the same solution as weighted least squares ($P = M$), weighted minimum length ($P = N$), or weighted generalized inverse approaches. If the probability density functions are not Gaussian, then maximum likelihood approaches can lead to different solutions. If the distributions are Gaussian, however, then all of the modifications introduced in Section 8.2 for the generalized inverse can be thought of as the maximum likelihood approach.

CHAPTER 9: CONTINUOUS INVERSE PROBLEMS AND OTHER APPROACHES

9.1 Introduction

Until this point, we have only considered discrete inverse problems, either linear or nonlinear, that can be expressed in the form

$$\mathbf{d} = \mathbf{G}\mathbf{m} \quad (1.13)$$

We now turn our attention to another branch on inverse problems, called continuous inverse problems, in which at least the model parameter vector \mathbf{m} is replaced by a continuous function, and the matrix \mathbf{G} is replaced by an integral relationship. The general case of a linear continuous inverse problem involving continuous data and a continuous model function is given by a Fredholm equation of the first kind

$$g(y) = \int k(x, y)m(x)dx \quad \text{continuous data } g(y) \quad (9.1)$$

where the data g are a continuous function of some variable y , the model m is a continuous function of some other variable x , and k , called the data kernel or Green's function, is a function of both x and y .

In most situations, data are not continuous, but rather are a finite sample. For example, analog (continuous) seismic data is digitized to become a finite, discrete data set. In the case where there are N data points, Equation (9.1) becomes

$$g_j = \int k_j(x)m(x)dx \quad \text{discrete data, } j = 1, N \quad (9.2)$$

One immediate implication of a finite data set of dimension N with a continuous (infinite dimensional) model is that the solution is underconstrained, and if there is any solution $m(x)$ that fits the data, there will be an infinite number of solutions that will fit the data as well. This basic problem of nonuniqueness was encountered before for the discrete problem in the minimum length environment (Chapter 3).

Given that all continuous inverse problems with discrete data are nonunique, and almost all real problems have discrete data, the goals of a continuous inverse analysis can be somewhat different than a typical discrete analysis. Three possible goals of a continuous analysis include: (1) find a model $m(x)$ that fits the data g_j , also known as *construction*, (2) find unique properties or values of all possible solutions that fit the data by taking linear combinations of the data, also known as *appraisal*, and (3) find the values of other linear combinations of the model using the data g_j , also known as *inference* (Oldenburg, 1984). There are many parallels, and some fundamental differences, between discrete and continuous inverse theory. For example, we have encountered the construction phase before for discrete problems whenever we used some operator

to find a solution that best fits the data. The appraisal phase is most similar to an analysis of resolution and stability analysis for discrete problems. We have not encountered the inference phase before. Emphasis in this chapter on continuous inverse problems will be on the construction and appraisal phases, and the references, especially Oldenburg [1984] can be used for further information on the inference phase.

The material in this chapter is based primarily on the following references:

Backus, G. E. and J. F. Gilbert, Numerical application of a formalism for geophysical inverse problems, *Geophys. J. Roy. Astron. Soc.*, 13, 247–276, 1967.

Huestis, S. P., An introduction to linear geophysical inverse theory, unpublished manuscript, 1992.

Jeffrey, W., and R. Rosner, On strategies for inverting remote sensing data, *Astrophys. J.*, 310, 463–472, 1986.

Oldenburg, D. W., An introduction to linear inverse theory, *IEEE Trans. Geos. Remote Sensing*, Vol. GE-22, No. 6, 665–674, 1984.

Parker, R. L., Understanding inverse theory, *Ann. Rev. Earth Planet. Sci.*, 5, 35–64, 1977.

Parker, R. L., *Geophysical Inverse Theory*, Princeton University Press, 1994.

9.2 The Backus–Gilbert Approach

There are a number of approaches to solving Equations (9.1) or (9.2). This chapter will deal exclusively with one approach, called the Backus–Gilbert approach, which was developed by geophysicists in the 1960's. This approach is based on taking a linear combination of the data g_j , given by

$$\begin{aligned}
 l &= \sum_{j=1}^N \alpha_j g_j = \sum_{j=1}^N \alpha_j \int k_j(x) m(x) dx \\
 &= \int \left[\sum_{j=1}^N \alpha_j k_j(x) \right] m(x) dx
 \end{aligned}
 \tag{9.3}$$

where the α_j are as yet undefined constants. The essence of the Backus-Gilbert approach is deciding how the α_j 's are chosen.

If the expression in square brackets, $[\sum \alpha_j k_j(x)]$, has certain special properties, it is possible in theory to construct a solution from the linear combination of the data. Specifically, if

$$\left[\sum_{j=1}^N \alpha_j k_j(x) \right] = \delta(x - x_0)
 \tag{9.4}$$

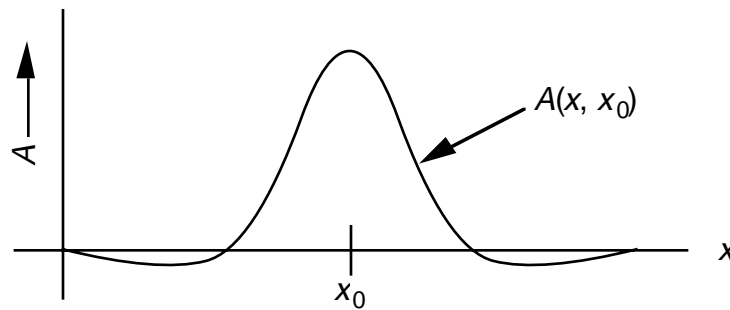
where $\delta(x - x_0)$ is the Dirac delta function at x_0 , then

$$l(x_0) = \left[\sum_{j=1}^N \alpha_j(x_0) g_j \right] = \int \delta(x - x_0) m(x) dx = m(x_0) \quad (9.5)$$

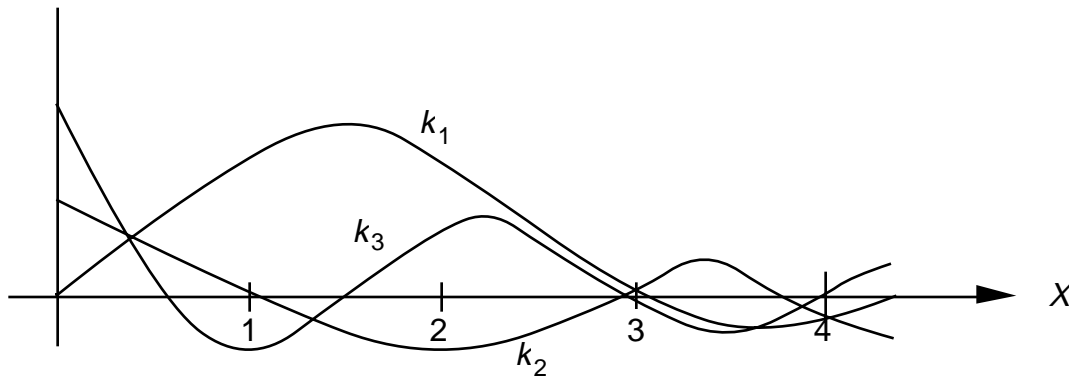
That is, choosing the α_j such that $[\sum \alpha_j k_j(x)]$ is as much like a delta function as possible at x_0 , we can obtain an estimate of the solution at x_0 . The expression $[\sum \alpha_j k_j(x)]$ plays such a fundamental role in Backus–Gilbert theory that we let

$$\sum_{j=1}^N \alpha_j k_j(x) = A(x, x_0) \quad (9.6)$$

where $A(x, x_0)$ has many names in the literature, including *averaging kernel*, *optimal averaging kernel*, *scanning function*, and *resolving kernel*. The averaging kernel $A(x, x_0)$ may take many different forms, but in general is a function which at least peaks near x_0 , as shown below



Recall from Equation (9.6) that the averaging kernel $A(x, x_0)$ is formed as a linear function of a finite set of data kernels k_j . An example of a set of three data kernels k_i over the interval $0 < x < 4$ is shown below



One of the fundamental problems encountered in the Backus-Gilbert approach is that the finite set of data kernels $k_j, j = 1, N$ is an incomplete set of basis functions from which to construct the Dirac delta function. You may recall from Fourier analysis, for example, that a spike (Dirac delta) function in the spatial domain has a white spectrum in the frequency domain, which implies that it takes an infinite sum of sin and cosine terms (basis functions) to construct the spike function. It is thus impossible for the averaging kernel $A(x, x_0)$ to exactly equal the Dirac delta with a finite set of data kernels k_j .

Much of Backus–Gilbert approach thus comes down to deciding how best to make $A(x, x_0)$ approach a delta function. Backus and Gilbert defined three measures of the “deltaness” of $A(x, x_0)$ as follows

$$J = \int [A(x, x_0) - \delta(x - x_0)]^2 dx \quad (9.7)$$

$$K = 12 \int (x - x_0)^2 [A(x, x_0) - \delta(x - x_0)]^2 dx \quad (9.8)$$

$$W = \int \left[H(x - x_0) - \int A(x, x_0) dx \right]^2 dx \quad (9.9)$$

where $H(x - x_0)$ is the Heaviside, or unit step, function at x_0 .

The smaller J , K , or W is, the more the averaging kernel approaches the delta function in some sense. Consider first the K criterion:

$$K = 12 \int \left\{ (x - x_0)^2 [A(x, x_0)]^2 - 2(x - x_0)^2 A(x, x_0) \delta(x - x_0) + (x - x_0)^2 [\delta(x - x_0)]^2 \right\} dx \quad (9.10)$$

The second and third terms drop out because $\delta(x - x_0)$ is nonzero only when $x = x_0$, and then the $(x - x_0)^2$ term is zero. Thus

$$\begin{aligned} K &= 12 \int (x - x_0)^2 [A(x, x_0)]^2 dx \\ &= 12 \int (x - x_0)^2 \left[\sum_{j=1}^N \alpha_j k_j(x) \right]^2 dx \end{aligned} \quad (9.11)$$

We minimize K by taking the partials with respect to the α_j 's and setting them to zero.

$$\begin{aligned} \frac{\partial K}{\partial \alpha_i} &= 12 \int \left\{ (x - x_0)^2 2 \left[\sum_{j=1}^N \alpha_j k_j(x) \right] k_i(x) \right\} dx = 0 \\ &= 24 \sum_{j=1}^N \alpha_j \int (x - x_0)^2 k_i(x) k_j(x) dx = 0 \end{aligned} \quad (9.12)$$

Writing out the sum over j explicitly for the i th partial derivative gives

$$\begin{aligned} \frac{\partial K}{\partial \alpha_i} = & \left[\int (x - x_0)^2 k_i(x) k_1(x) dx \right] \alpha_1 + \left[\int (x - x_0)^2 k_i(x) k_2(x) dx \right] \alpha_2 + \dots \\ & + \left[\int (x - x_0)^2 k_i(x) k_N(x) dx \right] \alpha_N = 0 \end{aligned} \quad (9.13)$$

Combining the N partial derivatives and using matrix notation this becomes

$$\begin{aligned} \begin{bmatrix} \int (x - x_0)^2 k_1 k_1 dx & \int (x - x_0)^2 k_1 k_2 dx & \dots & \int (x - x_0)^2 k_1 k_N dx \\ \int (x - x_0)^2 k_2 k_1 dx & \int (x - x_0)^2 k_2 k_2 dx & \dots & \int (x - x_0)^2 k_2 k_N dx \\ \vdots & \vdots & \ddots & \vdots \\ \int (x - x_0)^2 k_N k_1 dx & \int (x - x_0)^2 k_N k_2 dx & \dots & \int (x - x_0)^2 k_N k_N dx \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ N \times N & \hspace{15em} N \times 1 \quad N \times 1 \end{aligned} \quad (9.14)$$

or

$$B \boldsymbol{\alpha} = \mathbf{0} \quad (9.15)$$

$B \boldsymbol{\alpha} = \mathbf{0}$ has the trivial solution $\boldsymbol{\alpha} = \mathbf{0}$. Therefore, we add the constraint, with Lagrange multipliers, that

$$\int A(x, x_0) dx = 1 \quad (9.16)$$

which says the area under the averaging kernel is one, or that $A(x, x_0)$ is a unimodular function. Adding the constraint to the original K criterion creates a new criterion K' given by

$$K' = 12 \int (x - x_0)^2 [A(x, x_0)]^2 dx + \lambda \left\{ \left[\int A(x, x_0) dx \right] - 1 \right\} \quad (9.17)$$

which leads to

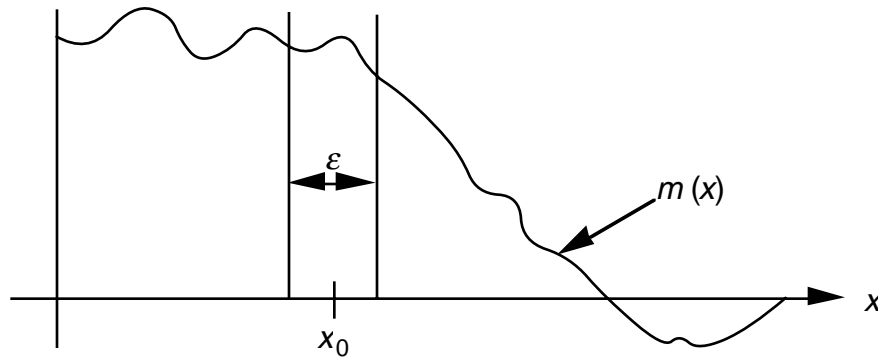
$$\begin{aligned} \begin{bmatrix} 24 \int (x - x_0)^2 k_1 k_1 dx & \dots & 24 \int (x - x_0)^2 k_1 k_N dx & \int k_1 dx \\ \vdots & \ddots & \vdots & \vdots \\ 24 \int (x - x_0)^2 k_N k_1 dx & \dots & 24 \int (x - x_0)^2 k_N k_N dx & \int k_N dx \\ \int k_1 dx & \dots & \int k_N dx & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ (N + 1) \times (N + 1) & \hspace{15em} \end{aligned} \quad (9.18)$$

or

$$C \begin{bmatrix} \alpha \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (9.19)$$

Then the α_j 's are found by inverting the square, symmetric matrix C .

The factor of 12 in the original definition of K' was added to facilitate the geometrical interpretation of K' . Specifically, with the factor of 12 included, $K' = \varepsilon$ if $A(x, x_0)$ is a unimodular (i.e., $\int A(x, x_0) dx = 1$) box car of width ε centered on x_0 . In general, when K' is small, it is found (Oldenburg, 1984, p 669) that K' is a good estimation of the *width* of the averaging function $A(x, x_0)$ at half its maximum value: "Thus K' gives essential information about the resolving power of the data." If K' is large, then the estimate of the solution $m(x)$ will be a smoothed average of the true solution around x_0 , and the solution will have poor resolution. Of course, you can also look directly at $A(x, x_0)$ and get much the same information. If $A(x, x_0)$ has a broad peak around x_0 , then the solution will be poorly resolved in that neighborhood. Features of the solution $m(x)$ with a scale length less than the width of $A(x, x_0)$ cannot be resolved (i.e., are nonunique). Thus, on the figure below, the high-frequency variations in $m(x)$ are not resolved.



The analysis of the averaging kernel above falls within the appraisal phase of the possible goals of a continuous inverse problem. Since any solution to Equation (9.2) is nonunique, often times the most important aspect of the inverse analysis is the appraisal phase.

The above discussion does not include possible data errors. Without data errors, inverting C to get the α_j 's gives you the solution. Even if C is nearly singular, then except for numerical instability, once you have the α_j 's, you have a solution. If, however, the data contain noise, then near-singularity of C can cause large errors in the solution for small fluctuations in data values.

It may help to think of the analogy between k_j and the k th row of \mathbf{G} in the discrete case $\mathbf{Gm} = \mathbf{d}$. Then

$$A(x, x_0) = \sum_{j=1}^N \alpha_j k_j$$

is equivalent to taking some linear combination of the rows of \mathbf{G} (which are M -dimensional, and therefore represent vectors in model space) and trying to make a row-vector as close as possible to a row of the identity matrix. If there is a near-linear dependency of rows in G , then the coefficients in the linear combination will get large. One can also speak of the near interdependence of the data kernels $k_j(x)$, $j = 1, N$ in Hilbert space. If this is the case, then terms like $\int k_i k_j dx$ can be large, and C will be nearly singular. This will lead to *large values for α* as it tries to approximate $\delta(x - x_0)$ with a set of basis functions (kernels) k_j that have near interdependence. Another example arises in the

figure after Equation (9) with three data kernels that are all nearly zero at $x = 3$. It is difficult to approximate a delta function near $x = 3$ with a linear combination of the data kernels, and the coefficients of that linear combination are likely to be quite large.

You can quantify the effect of the near singularity of C by considering the data to have some covariance matrix [cov \mathbf{d}]. Then the variance of your estimate of the solution $m(x_0)$ at x_0 is

$$\sigma_{m(x_0)}^2 = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_N] [\text{cov } \mathbf{d}] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \quad (9.20)$$

and if

$$[\text{cov } \mathbf{d}] = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_N^2 \end{bmatrix}$$

then

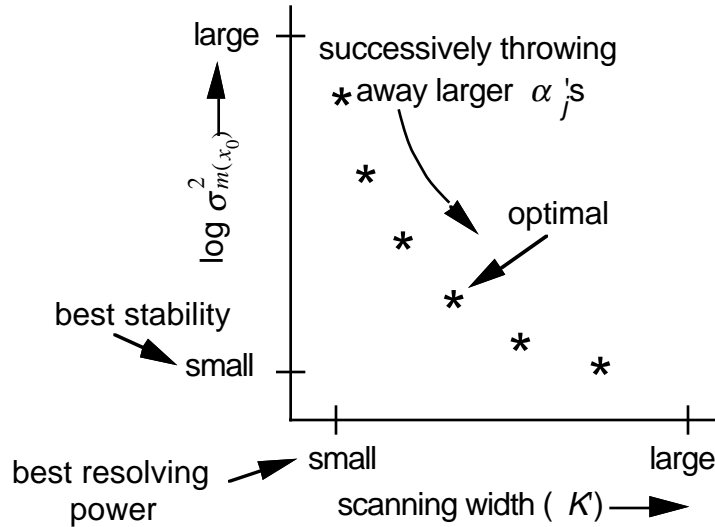
$$\sigma_{m(x_0)}^2 = \sum_{j=1}^N \alpha_j^2 \sigma_j^2 \quad (9.21)$$

[See Oldenburg, 1984, Equation (18), p. 670, or Jeffrey and Rosner, Equation (3.5), p. 467.]

If $\sigma_{m(x_0)}$ is large, the solution is unstable. We now have two conflicting goals:

- versus
- (1) minimize K'
 - (2) minimize $\sigma_{m(x_0)}$

This leads to *trade-off* curves of stability (small $\sigma_{m(x_0)}$) versus resolving power (small K'). These trade-off curves are typically plotted as (next page)



Typically, one gains a lot of stability without too much loss in resolving power early on by throwing away the largest (few) α_j 's.

Another way to accomplish the same goal is called *spectral expansion* or *spectral synthesis*. With this technique, you do singular-value decomposition on C and start throwing away the smallest singular values and associated eigenvectors until the error amplification (i.e., $\log \sigma_{m(x_0)}$) is sufficiently small. In either case, you give up resolving power to gain stability. Small singular values, or large α_j , are associated with high-frequency components of the solution $m(x)$. As you give up resolving power to gain stability, the “width” of the resolving kernel increases. Thus, in general, the narrower the resolving kernel, the better the resolution but the poorer the stability. Similarly, the wider the resolving kernel, the poorer the resolution, but the better the stability.

The logic behind the trade-off between resolution and stability can be looked at another way. With the best resolving power you obtain the best fit to the data in the sense of minimizing the difference between observed and predicted data. However, if the data are known to contain noise, then fitting the data exactly (or too well) would imply fitting the noise. It does not make sense to fit the data better than the data uncertainties in this case. We can quantify this relation for the case in which the data errors are Gaussian and uncorrelated with standard deviation σ_j by introducing χ^2

$$\chi^2 = \sum_{j=1}^N (g_j - \hat{g}_j)^2 / \sigma_j^2 \tag{9.22}$$

where g_j is the predicted j th datum, which depends on the choice of α . χ^2 will be in the range $0 \leq \chi^2 \leq \infty$. If $\chi^2 = 0$, then the data are fit perfectly, noise included. If $\chi^2 \approx N$, then the data are being fit at about the one standard deviation level. If $\chi^2 \gg N$, then the data are fit poorly. By using the trade-off curve, or the spectral expansion method, you affect the choice of α_j 's, and hence the predicted data g_j and ultimately the value of χ^2 . Thus the best solution is obtained by adjusting the α_j until $\chi^2 \approx N$.

Now reconsider the J criterion:

$$\begin{aligned}
 J &= \int [A(x, x_0) - \delta(x - x_0)]^2 dx \\
 &= \int [A^2(x, x_0) - 2A(x, x_0)\delta(x - x_0) + \delta^2(x - x_0)] dx
 \end{aligned} \tag{9.23}$$

Recall that for the K criterion, the second and third terms vanished because they were multiplied by $(x - x_0)^2$, which is zero at the only place the other two terms are nonzero because of the $\delta(x - x_0)$ term. With the J criterion, it is minimized by taking partial derivatives with respect to α_i and setting equal to zero. The $A^2(x, x_0)$ terms are similar to the K criterion case, but

$$\begin{aligned}
 \frac{\partial}{\partial \alpha_i} \left[\int -2A(x, x_0)\delta(x - x_0) \right] &= -2 \frac{\partial}{\partial \alpha_i} \int \sum_{j=1}^N \alpha_j k_j(x) \delta(x - x_0) dx \\
 &= -2 \int k_i(x) \delta(x - x_0) dx = -2k_i(x_0)
 \end{aligned} \tag{9.24}$$

Thus the matrix equations from minimizing J become

$$\begin{matrix}
 \begin{bmatrix} \int k_1 k_1 & \int k_1 k_2 & \cdots & \int k_1 k_N \\ \int k_1 k_2 & \int k_2 k_2 & \cdots & \int k_2 k_N \\ \vdots & \vdots & \ddots & \vdots \\ \int k_1 k_N & \int k_2 k_N & \cdots & \int k_N k_N \end{bmatrix} & \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} & = & 2 \begin{bmatrix} k_1(x_0) \\ k_2(x_0) \\ \vdots \\ k_N(x_0) \end{bmatrix} \\
 N \times N & N \times 1 & & N \times 1
 \end{matrix} \tag{9.25}$$

or

$$\mathbf{D} \boldsymbol{\alpha} = 2\mathbf{k} \tag{9.26}$$

Notice now that $\boldsymbol{\alpha} = \mathbf{0}$ is not a trivial solution, as it was in the case of the K criterion. Thus, we do not have to use Lagrange multipliers to add a constraint to insure a nontrivial solution. Also, notice that \mathbf{D} , the matrix that must be inverted to find $\boldsymbol{\alpha}$, no longer depends on x_0 , and thus need only be formed once. This benefit is often sacrificed by adding the constraint that $A(x, x_0)$ be unimodular anyway! Then the $\boldsymbol{\alpha}$'s found no longer necessarily give $m(x_0)$ corresponding to a solution of $g_j = \int k_i(x)m(x) dx$ at all, but this may be acceptable if your primary goal is appraisal of the properties of solutions, rather than construction of one among many possible solutions.

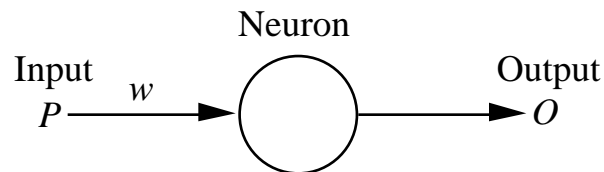
The J and K' criteria lead to different $\boldsymbol{\alpha}$. The J criterion typically leads to narrower resolving kernels, but often at the expense of negative side lobes. These side lobes confuse the interpretation of $A(x, x_0)$ as an "averaging" kernel. Thus, most often the K' criterion is used, even though it leads to somewhat broader resolving kernels.

9.3 Neural Networks

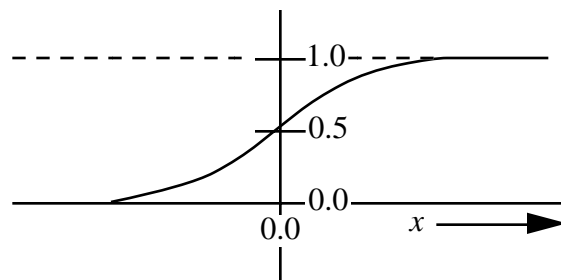
Neural networks, based loosely on models of the brain and dating from research in the 1940s (long before the advent of computers), have become very successful in a wide range of applications, from pattern recognition (one of the earliest applications) to aerospace (autopilots, aircraft control systems), defense (weapon steering, signal processing), entertainment (animation and other special effects), manufacturing (quality control, product design), medical (breast cancer cell analysis, EEG analysis, optimization of transplant times), oil and gas (exploration), telecommunications (image and data compression), and speech (speech recognition) applications.

The basic idea behind neural networks is to design a system, based heavily on a parallel processing architecture, that can learn to solve problems of interest.

We begin our discussion of neural networks by introducing the neuron model, which predictably has a number of names, including, of course, *neurons*, but also *nodes*, *units*, and *processing elements* (PEs)



where the neuron sees some input P coming, which is weighted by w before entering the neuron. The neuron processes this weighted input $= Pw$ and creates an output O . The output of the neuron can take many forms. In early models, the output was always $+1$ or -1 because the neural network was being used for pattern recognition. Today, this neuron model is still used (called the step function, or Heaviside function, among other things), but other neuron models have the output equal to the weighted input to the neuron (called the linear model) and perhaps the most common of all, the sigmoid model, which looks like



often taken to be given by $S(x) = 1/(1 + e^{-x})$, where x is the weighted input to the neuron. This model has elements of both the linear and step function but has the advantage over the step function of being continuously differentiable.

So, how is the neuron model useful? It is useful because, like the brain, the neuron can be trained and can learn from experience. What it learns is w , the correct weight to give the input so that the output of the neuron matches some desired value.

As an almost trivial example, let us assume that our neuron in the figure above behaves as a linear model, with the output equal to the weighted input. We train the neuron to learn the correct w by giving it examples of inputs and output that are true. For example, consider examples to be

input =	1	output =	2
	10		20
	-20		-40
	42		84

By inspection we recognize that $w = 2$ is the correct solution. However, in general we start out not knowing what w should be. We thus begin by assuming it to be some number, perhaps 0, or as is typically done, some random number. Let us assume that $w = 0$ for our example. Then for the first test with input = 1, our output would be 0. The network recognizes that the output does not match the desired output and changes w . There are a world of ways to change w based on the mismatch between the output and the desired output (more on this later), but let us assume that the system will increase w , but not all the way to 2, stopping at 0.5. Typically, neural networks do not change w to perfectly fit the desired output because making large changes to w very often results in instability. Now our system, with $w = 0.5$, inputs 10 and output 5. It again increases w and moves on to the other examples. When it has cycled through the known input/output pairs once, this is called an *epoch*. Once it has gone through an epoch, it goes back to the first example and cycles again until there is an acceptably small misfit between all of the outputs and desired outputs for all of the known examples. In neural network programs, the codes typically go through all of the known examples randomly rather than sequentially, but the idea is the same. In our example, it will settle in eventually on $w = 2$. In more realistic examples, it takes hundreds to thousands of iterations (one iteration equals an epoch) to find an acceptable set of weights.

In our nomenclature, we have the system

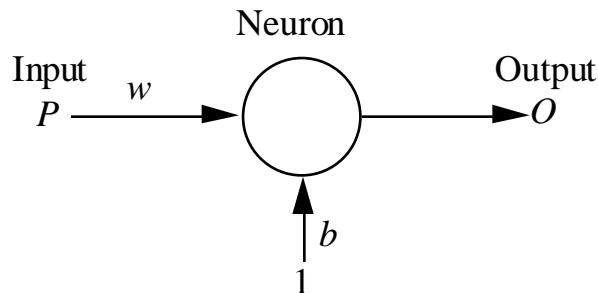
$$\mathbf{d} = \mathbf{Gm}$$

and in this example, $G = 2$. The beauty of the neural network is that it “learned” this relationship without even having to know it formally. It did it simply by adjusting weights until it was able to correctly match a set of example input/output pairs.

Suppose we change our example input/output pairs to

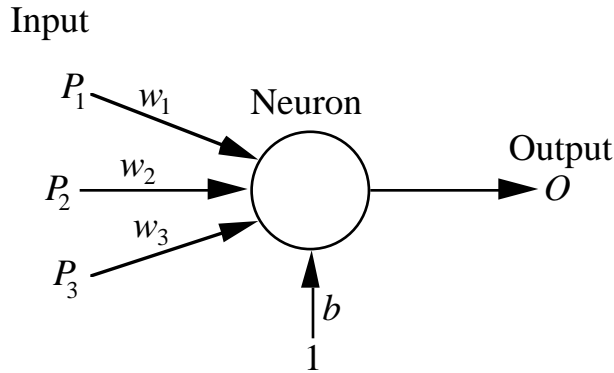
input =	1	output =	5
	10		23
	-20		-37
	42		87

where the output has been shifted 3 units from the previous example. You may recognize that we will be unable to find a single weight w to make the output of our neuron equal the desired output. This leads to the first additional element in the neuron model, called the *bias*. Consider the diagram below



where the bias is something added to the weighted input Pw before the neuron “processes” it to make output. It is convention that the bias has an input of 1 and another “weight” (bias) b that is to be determined in the learning process. In this example, the neuron will “learn” that, as before, $w = 2$, and now that the bias b is 3.

The next improvement to our model is to consider a neuron that has multiple inputs:

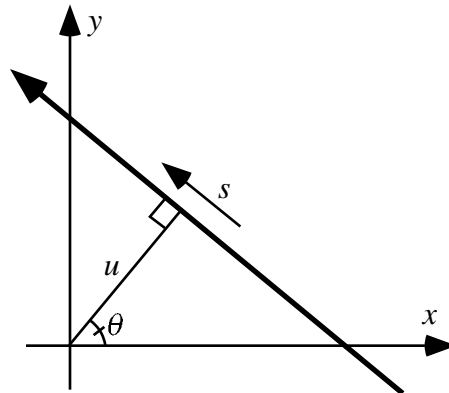


We now introduce the nomenclature that the process of the neuron is going to be some function of the sum of the weighted input vector and the bias b . That is, we describe the neuron by the function of $(\mathbf{P} \cdot \mathbf{w} + b)$, where

$$\mathbf{P} \cdot \mathbf{w} = P_1w_1 + P_2w_2 + \dots + P_Nw_N = \sum_{i=1}^N P_iw_i \quad (9.27)$$

This is a very incomplete introduction to neural networks. Perhaps someday we will add more material.

9.4 The Radon Transform and Tomography (Approach 1)



9.4.1 Introduction

Consider a model space $m(x, y)$ through which a straight-line ray is parameterized by the perpendicular distance u and angle θ . Position (x, y) and ray coordinates (u, s) are related by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u \\ s \end{bmatrix} \quad (9.28)$$

and

$$\begin{bmatrix} u \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (11.18 \text{ in Mencke})$$

If $m(x, y)$ is slowness ($1/v$) model, and $t(u, \theta)$ is the travel time along a ray at distance u and angle θ , then

$$t(u, \theta) = \int_{-\infty}^{\infty} m(x, y) ds \quad (9.29)$$

is known as the Radon transform (RT). Another way of stating this is that $t(u, \theta)$ is the "projection" of $m(x, y)$ onto the line defined by u and θ .

The inverse problem is: Given $t(u, \theta)$ for many values of u and θ , find the model $m(x, y)$, i.e., the inverse Radon transform (IRT).

Define a 2-D Fourier transform of the model:

$$\tilde{m}(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x, y) e^{-2i\pi(k_x x + k_y y)} dx dy \quad (9.30)$$

$$m(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{m}(k_x, k_y) e^{2i\pi(k_x x + k_y y)} dk_x dk_y \quad (9.31)$$

Now, define the 1-D Fourier Transform of the projection data:

$$\tilde{t}(k_u, \theta) = \int_{-\infty}^{\infty} t(u, \theta) e^{-2i\pi k_u u} du \quad (9.32)$$

$$t(u, \theta) = \int_{-\infty}^{\infty} \tilde{t}(k_u, \theta) e^{2i\pi k_u u} dk_u \quad (9.33)$$

Substituting Equation (1) into Equation (4) gives

$$\tilde{t}(k_u, \theta) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} m(x, y) ds \right] e^{-2i\pi k_u u} du \quad (9.34)$$

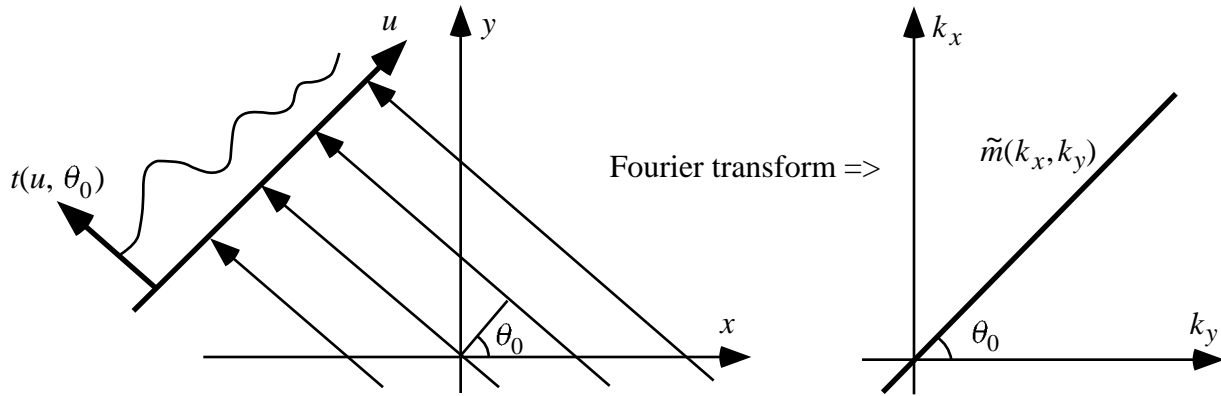
Making a change of variables $ds du \rightarrow dx dy$ and using the fact that the Jacobian determinant is unity we have

$$\begin{aligned} \tilde{t}(k_u, \theta) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x, y) e^{-2i\pi k_u (\cos \theta x + \sin \theta y)} dx dy \\ &= \tilde{m}(k_u \cos \theta, k_u \sin \theta) \end{aligned} \quad (9.35)$$

Equation (9.35) states that the 1-D Fourier transform of the projected data is equal to the 2-D Fourier transform of the model. This relationship is known as the Fourier (central) slice

theorem because for a fixed angle θ , the projected data provide the Fourier transform of the model slice through the origin of the wavenumber space, i.e.,

$$\begin{bmatrix} k_x \\ k_y \end{bmatrix} = k_u \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix}, \theta_0 \text{ fixed}$$



Now, we can invert the 2-D Fourier transform and recover the model,

$$m(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{m}[k_u(x \cos \theta + y \sin \theta)] e^{2i\pi(k_x x + k_y y)} dk_x dk_y \quad (9.36)$$

Change variables to polar coordinates gives

$$m(x, y) = \int_{-\infty}^{\infty} \int_0^{\pi} \tilde{m}(k_u, \theta) e^{2i\pi k_u(x \cos \theta + y \sin \theta)} |k_u| d\theta dk_u \quad (9.37)$$

where $|k|$ arises because $k|_{-\infty}^{\infty} \rightarrow d\theta|_0^{\pi}$.

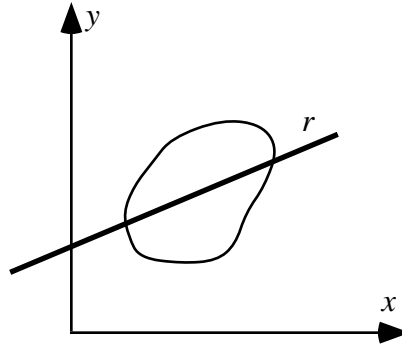
$$\begin{aligned} m(x, y) &= \int_0^{\pi} \int_{-\infty}^{\infty} \tilde{m}(k_u, \theta) |k| e^{2i\pi k_u(x \cos \theta + y \sin \theta)} dk_u d\theta \\ &= \int_0^{\pi} m^*(k_u(x \cos \theta + y \sin \theta), \theta) d\theta \end{aligned} \quad (9.38)$$

where m^* is obtained from the inverse Fourier transform of $|k_u| \tilde{m}(k_u, \theta)$.

The Radon Transform can be written more simply as

$$F(a, b) = \int_r f(x, a + bx) dr \quad (9.39)$$

shadow path model

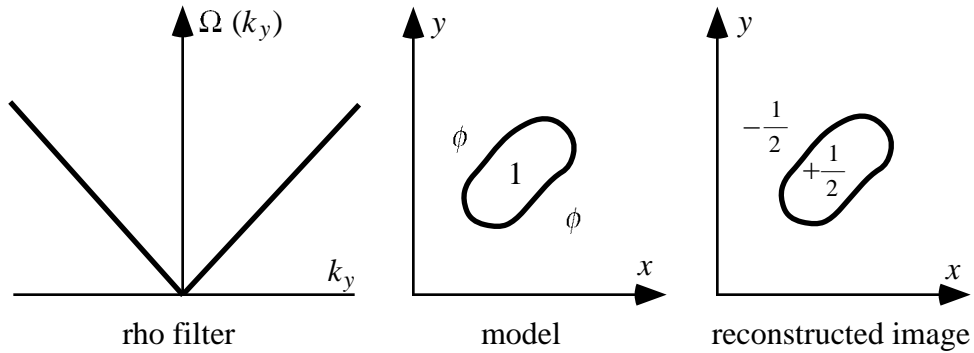


with the inverse

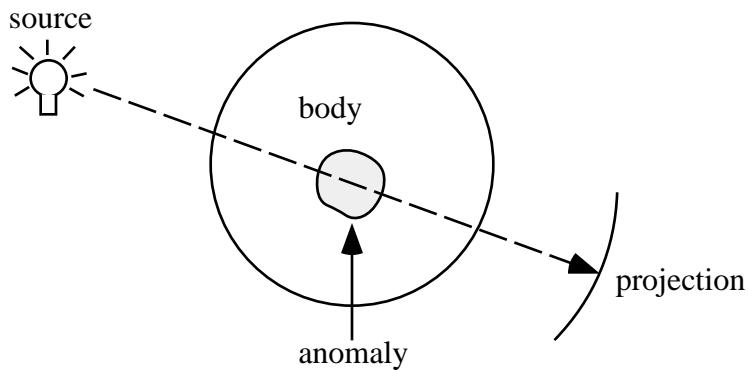
$$f(x, y) = \Omega(x, y) * \int_q F(y - bx, b) dq \quad (9.40)$$

model "rho filter" path shadows

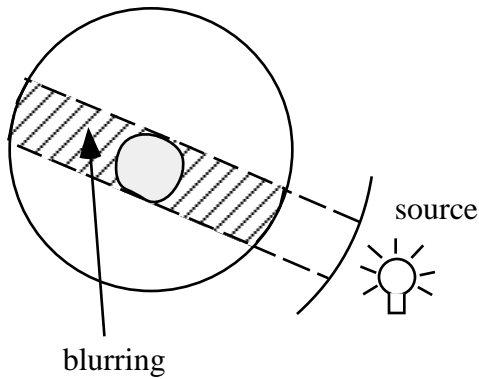
Note the similarity of the inverse and forward transforming with a change of sign in the argument of the integrand.



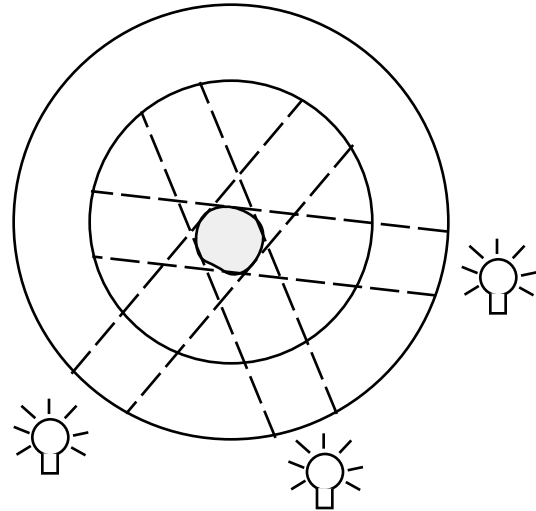
9.4.2 Interpretation of Tomography Using the Radon Transform



The resolution and accuracy of the reconstructed image are controlled by the coverage of the sources.



Back projection with one ray

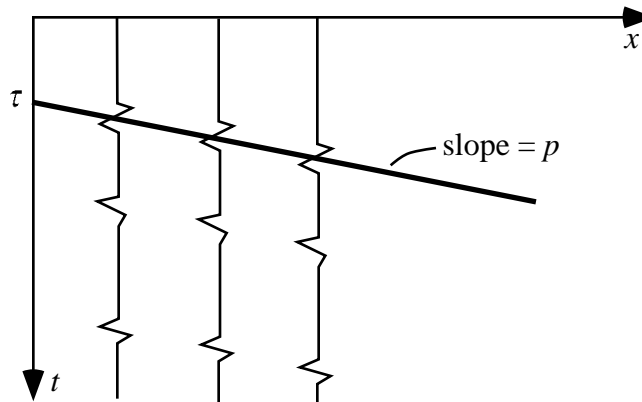


Back projection with many rays

Because of this requirement for excellent coverage for a good reconstruction, the Radon transform approach is generally not used much in geophysics. One exception to this is in the area of seismic processing of petroleum industry data, where data density and coverage is, in general, much better.

9.4.3 Slant-Stacking as a Radon Transform (following Claerbout, 1985)

Let $u(x, t)$ be a wavefield. An example would be as follows.

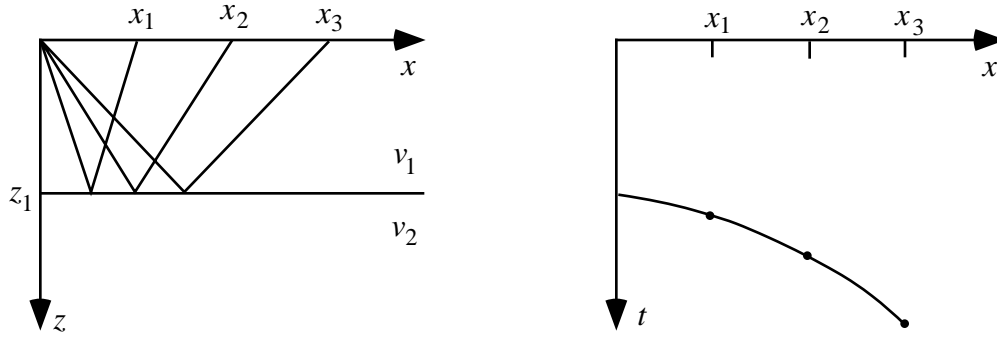


The *slant stack* of the wavefield is defined by

$$\bar{u}(p, \tau) = \int u(x, \tau + px) dx \tag{9.41}$$

The integral along x is done at constant τ , which defines a slanting straight line in the $x-t$ plane with slope p . Note the similarity of Equation (9.41) to Equation (9.39). Equation (9.41) is a Radon transform.

To get a better “feel” for this concept, let's step back a little and consider the travel time equations of “rays” in a layered medium.



The travel time equation is

$$t = \sqrt{4z_1^2 + x^2} / v \quad \text{or} \quad t^2 v^2 - x^2 = 4z_1^2 \quad (9.42)$$

Because the signs of the t^2 and x^2 terms are opposite, this is an equation of a hyperbola. Slant stacking is changing variables from $t - x$ to $\tau - p$ where

$$\tau = t - px \quad (9.43)$$

and

$$p = \frac{dt}{dx} \quad (9.44)$$

From Equation (9.42) (?)

$$2t \, dt \, v^2 = 2x \, dx$$

so

$$\frac{dt}{dx} = \frac{x}{v^2 t} \quad (9.45)$$

The equations simplify if we introduce the parametric substitution.

$$\left. \begin{aligned} z &= vt \cos \theta \\ x &= vt \sin \theta \\ \text{so } x &= z \tan \theta \end{aligned} \right\} \quad (9.46)$$

Now, take the linear moveout Equation (9.43) and substitute for t and x (using $p = (\sin \theta) / v$).

$$\begin{aligned} \tau &= \frac{z}{v \cos \theta} - \frac{\sin \theta}{v} z \tan \theta \\ &= \frac{z}{v} \cos \theta \end{aligned} \quad (9.47)$$

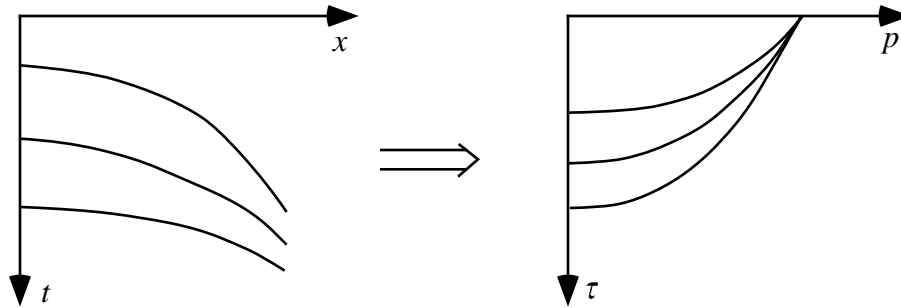
This can be written after some substitution as

$$\tau = \frac{z}{v} \sqrt{1 - p^2 v^2} \quad (9.48)$$

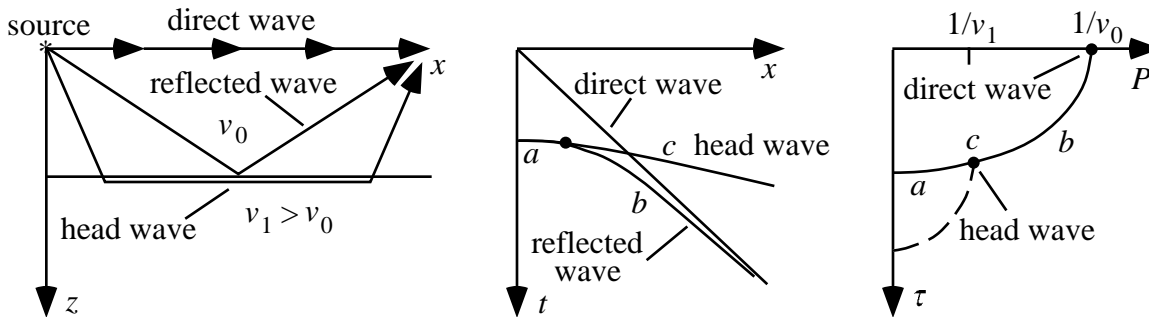
and finally,

$$\left(\frac{\tau}{x}\right)^2 + p^2 = \frac{1}{v^2} \tag{9.49}$$

This is an equation of an ellipse in τ - p space. All this was to show that hyperbolic curves in t - x space transform to ellipses in τ - p space.



With this background, consider how the wavefield in a layer-over-halfspace slant stacks (or Radon transforms).



A couple of important features of the τ - p plot are as follows:

1. Curves cross one another in t - x space but not in τ - p space.
2. The p axis has dimensions of $1/v$.
3. The velocity of each layer can be read from its τ - p curve as the inverse of the maximum p value on its ellipse.
4. Head waves are points in τ - p space located where the ellipsoids touch.

Returning to our original slant-stack equation,

$$\bar{u}(p, \tau) = \int u(x, \tau + px) dx \tag{9.41}$$

This equation can be transformed into the Fourier space using

$$U(k, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, t) e^{i\omega t - ikx} dx dt \tag{9.50}$$

Let $p = k / w$

$$U(wp, w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, t) e^{iw(t-px)} dx dt \quad (9.51)$$

and change of variables from t to $\tau = t - px$.

$$U(wp, w) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} u(x, t + px) dx \right] e^{iw\tau} d\tau \quad (9.52)$$

|<---(9.41)-->|

Insert Equation (9.41) to get

$$U(wp, w) = \int_{-\infty}^{\infty} \bar{u}(p, \tau) e^{iw\tau} d\tau \quad (9.53)$$

Think of this as a 1-D function of w that is extracted from the k - w plane along the line $k = wp$.

Finally, taking the inverse Fourier transform

$$\bar{u}(p, \tau) = \int_{-\infty}^{\infty} U(wp, w) e^{-iw\tau} dw \quad (9.54)$$

This result shows that a slant stack can be done by Fourier transform operations.

1. Fourier transform $u(x, t)$ to $U(k, w)$.
2. Extract $U(wp, w)$ from $U(k, w)$ along the line $k = wp$.
3. Inverse Fourier transform from w to t .
4. Repeat for all interesting values of p .

Tomography is based on the same mathematics as inverse slant stacking. In simple terms, tomography is the reconstruction of a function given line integrals through it.

The inverse slant stack is based on the inverse Fourier transform of Equation (9.50),

$$u(x, t) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} U(k, w) e^{ikx} dk \right] e^{-iwt} dw \quad (9.55)$$

substituting $k = wp$ and $dk = wpd$. Notice that when w is negative, the integration with dp is from positive to negative. To keep the integration in the conventional sense, we introduce $|w|$,

$$u(x, t) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} U(wp, w) |w| e^{iwp x} dp \right] e^{-iwt} dw \quad (9.56)$$

Changing the order of integration gives

$$u(x, t) = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} [U(wp, w) e^{iwp x} |w|] e^{-iwt} dw \right\} e^{-iwt} dp \quad (9.57)$$

Note that the term in { } contains the inverse Fourier transform of a product of three functions of frequency. The three functions are (1) $U(wp, w)$, which is the Fourier transform of the slant stack, (2) $e^{iwp x}$, which can be thought of as a delay operator, and (3) $|w|$, which is called the “rho” filter. A product of three terms in Fourier space is a convolution in the original space. Let the delay px be a time shift in the argument, so,

$$u(x, t) = \text{rho}(t) * \int \bar{u}(p, t - px) dp \quad (9.58)$$

This is the inverse slant-stack equation. Comparing the forward and inverse Equations (9.41) and (9.58), note that the inverse is basically another slant stack with a sign change.

$$\bar{u}(p, \tau) = \int u(x, \tau + px) dx \quad (9.41)$$

$$u(x, t) = \text{rho}(t) * \int \bar{u}(p, t - px) dp \quad (9.58)$$

9.5 Review of the Radon Transform (Approach 2)

If $m(x, y)$ is a 2-D slowness ($1/v$) model, and $t(u, \theta)$ is the travel time along a ray parameterized by distance u and angle θ , then

$$t(u, \theta) = \int_{-\infty}^{\infty} m(x, y) ds \quad \text{is the Radon transform} \quad (9.59)$$

The inverse problem is: Given $t(u, \theta)$ for many values of u and θ , find the model $m(x, y)$. Take the 1-D Fourier transform of the projection data,

$$\tilde{t}(k_u, \theta) = \int_{-\infty}^{\infty} t(u, \theta) e^{-2i\pi k_u u} du \quad (9.60)$$

Substitute Equation (9.59),

$$\tilde{t}(k_u, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x, y) ds e^{-2i\pi k_u u} du \quad (9.61)$$

Change variables $ds du \rightarrow dx dy$

$$\tilde{t}(k_u, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x, y) e^{-2i\pi k_u (\cos x + \sin \theta y)} dx dy \quad (9.62)$$

Recognize that the right-hand side is a 2-D Fourier transform:

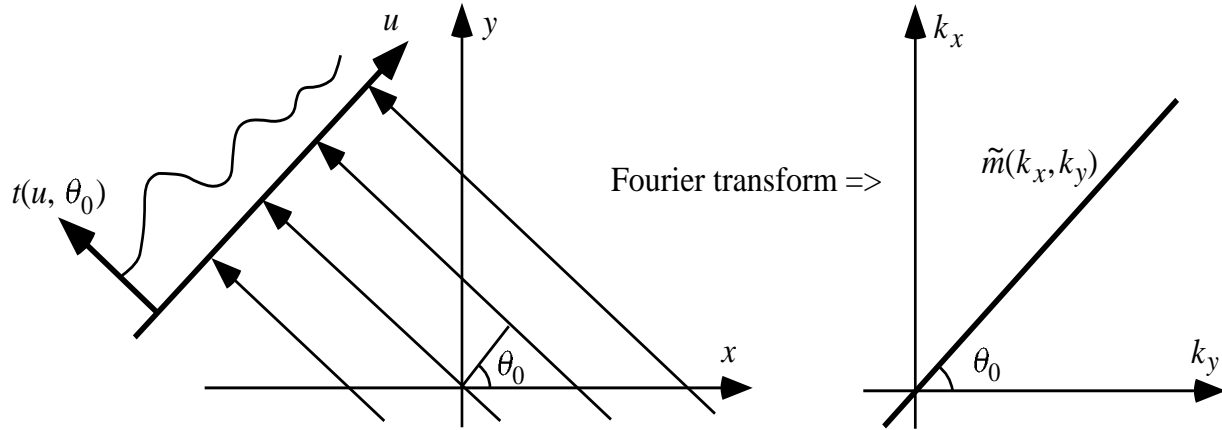
$$\tilde{m}(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(x, y) e^{-2i\pi(k_x y + k_y x)} dx dy \quad (9.63)$$

So

$\tilde{t}(k_u, \theta) = \tilde{m}(k_u \cos \theta, k_u \sin \theta)$

(9.64)

In words: “The 1-D Fourier transform of the projected data is equal to the 2-D Fourier transform of the model evaluated along a radial line in the k_x - k_y space with angle θ .”

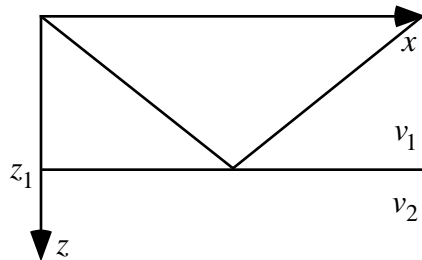


If the Radon transform is known for all values of (u, θ) , then the Fourier transform image of m is known for all values of (u, θ) . The model $m(x, y)$ can then be found by taking the inverse Fourier transform of $m(k_x, k_y)$.

Slant-stacking is also a Radon transform. Let $u(x, t)$ be a wavefield. Then the slant-stack is

$$\bar{u}(p, \tau) = \int u(x, \tau + px) dx \quad (9.65)$$

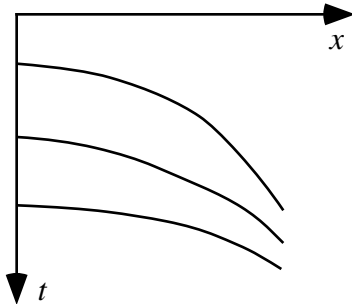
Now, consider a simple example of reflections in a layer.



Travel time ($t-x$) equation is:

$$t^2 v^2 - x^2 = z^2$$

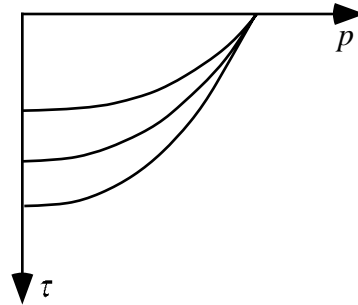
hyperbolas



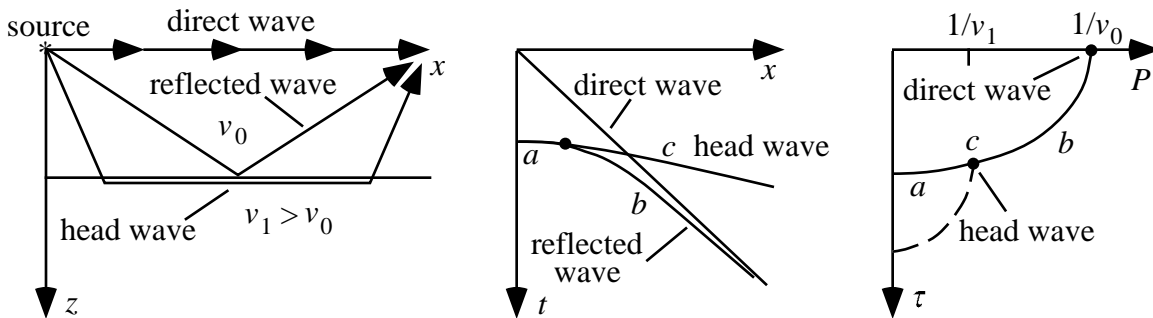
Slant-stack ($p-\tau$) equation is:

$$(\tau / x)^2 + p^2 = 1 / v^2$$

ellipses



With this background, consider how the “complete” wavefield in a layer slant-stack:



The $p-\tau$ equation can be transformed into $k-w$ space by a 2-D Fourier transform:

$$U(k, w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, t) e^{iwt - ikx} dx dt \tag{9.66}$$

The inverse slant-stack (IRT) is based on the inverse Fourier transform of the above equation,

$$u(x, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(k, w) e^{ikx} dx e^{-iwt} dw \tag{9.67}$$

Substitute $k = wp$ and $dk = w dp$. Notice that when w is negative, the integral with respect to dp is from + to -. To keep the integration in the conventional sense, introduce $|w|$.

$$u(x, t) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} U(wp, w) |w| e^{iwp x} dp \right] e^{-iwt} dw \tag{9.68}$$

Changing the order of integration,

$$u(x, t) = \int_{-\infty}^{\infty} \left\{ \left[\int_{-\infty}^{\infty} U(wp, w) e^{iwp x} |w| \right] e^{-iwt} dw \right\} dp \quad (9.69)$$

The term in { } contains an inverse Fourier transform of a product of three functions of w .

1. $U(wp, w)$ is the Fourier transform of the slant-stack evaluated at $k = wp$. So, the slant-stack is given by

$$\bar{u}(p, \tau) = \int_{-\infty}^{\infty} e^{-i w \tau} U(wp, w) dw \quad (9.70)$$

2. The $e^{iwp x}$ term can be thought of a delay operator where $\tau = t - px$.
3. The $|w|$ term is called the “rho” filter.

Now we can rewrite the above equation as

$$u(x, t) = \text{rho}(t) * \int \bar{u}(p, t - px) dp \quad \text{Inverse Radon Transform} \quad (9.71)$$

Compare with

$$\bar{u}(x, \tau) = \int u(x, \tau + px) dx \quad \text{Radon Transform} \quad (9.41)$$

9.6 Alternative Approach to Tomography

An alternative approach to tomography is to discretize the travel time equation, as follows.



$$m(x, y) = \frac{1}{v(x, y)} \quad (9.72)$$

$$t = \int_r m dl \quad (9.73)$$

With some assumptions, we can linearize and discretize this equation to

$$t_r = \sum_b l_{rb} m_b \quad (9.74)$$

- where
- t_r = travel time of the r^{th} ray
 - m_b = slowness of the b^{th} block
 - l_{rb} = length of the r^{th} ray segment in the b^{th} block.

In matrix form,

$$\mathbf{d} = \mathbf{Gm} \quad \text{Aha!!} \quad (9.75)$$

We know how to solve this equation. But what if we have a 3-D object with 100 blocks on a side? Then $M \approx (100)^3 = 10^6$, and even $\mathbf{G}^T\mathbf{G}$ is a matrix with M^2 elements, or $\sim 10^{12}$. Try throwing that into your MATLAB program. So what can we do?

Let's start at the beginning again.

$$\mathbf{d} = \mathbf{Gm} \quad (9.75)$$

$$\mathbf{G}^T\mathbf{d} = \mathbf{G}^T\mathbf{Gm} \quad (9.76)$$

$$\quad \quad \quad | \mathbf{R} |$$

In a sense, \mathbf{G}^T is an approximate inverse operator that transforms a data vector into model space. Also, in this respect, $\mathbf{G}^T\mathbf{G}$ can be thought of as a resolution matrix that shows you the "filter" between your estimate of the model, $\mathbf{G}^T\mathbf{d}$, and the real model, \mathbf{m} . Since the ideal $\mathbf{R} = \mathbf{I}$, let's try inverting Equation (9.76) by using only the diagonal elements of $\mathbf{G}^T\mathbf{G}$. Then the solution can be computed simply.

$$\begin{bmatrix} \sum_{i=1}^N l_{i_1} t_i \\ \sum_{i=1}^N l_{i_2} t_i \\ \vdots \\ \sum_{i=1}^N l_{i_M} t_i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N l_{i_1}^2 & 0 & 0 & 0 \\ 0 & \sum_{i=1}^N l_{i_2}^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sum_{i=1}^N l_{i_M}^2 \end{bmatrix} \mathbf{m} \quad (9.77)$$

so

$$m_b = \frac{\sum_{i=1}^N l_{ib} t_i}{\sum_{i=1}^N l_{i_b}^2} \quad \text{"tomographic approximation"} \quad (9.78)$$

Operationally, each ray is back-projected, and at each block a ray hits, the contributions to the two sums are accumulated in two vectors. At the end, the contents of each element of the numerator vector are divided by each element of the denominator vector. This reduces storage requirements to $\sim 2M$ instead of M^2 .

Let's see how this works for our simple tomography problem.

for $\mathbf{m} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ $\mathbf{d} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ (9.79)

$$\mathbf{m}_{ML} = \begin{bmatrix} \frac{3}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ -\frac{1}{4} \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \mathbf{G}^T \mathbf{d} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (9.80)$$

Fits data

$$\mathbf{G}^T \mathbf{G} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \quad [\mathbf{G}^T \mathbf{G}]_d = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad (9.81)$$

The approximate equation is

$$\begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \mathbf{m} \Rightarrow \tilde{\mathbf{m}} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \quad (9.82)$$

Not bad, but this does not predict the data. Can we improve on this? Yes! Following Menke, develop an iterative solution.

$$[\mathbf{G}^T \mathbf{G}] \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (9.83)$$

$$[\mathbf{I} - \mathbf{I} + \mathbf{G}^T \mathbf{G}] \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (9.84)$$

$$\mathbf{m} - [\mathbf{I} - \mathbf{G}^T \mathbf{G}] \mathbf{m} = \mathbf{G}^T \mathbf{d} \quad (9.85)$$

so

$$\mathbf{m}^{\text{est}(i)} = \mathbf{G}^T \mathbf{d} + [\mathbf{I} - \mathbf{G}^T \mathbf{G}] \mathbf{m}^{\text{est}(i-1)} \quad (9.86)$$

[Menke's Equations (11) and (24)]

A slightly different iterative technique,

$$\mathbf{m}^{(k)} = \mathbf{D}^{-1} \mathbf{G}^T \mathbf{d} + [\mathbf{D} - \mathbf{G}^T \mathbf{G}] \mathbf{m}^{(k-1)} \quad (9.87)$$

where \mathbf{D} is the diagonalized $\mathbf{G}^T \mathbf{G}$ matrix. Then for $\mathbf{m}^0 = 0$, $\mathbf{m}^1 = \mathbf{D}^{-1} \mathbf{G}^T \mathbf{d}$ is the “tomographic approximation” solution.

These are still relatively “rudimentary” iterative techniques. They do not always converge. Ideally, we want iterative techniques to converge to the least squares solution. A number of such techniques exist and have been used in “tomography.” They include the “simultaneous iterative reconstruction techniques,” or SIRT, and LSQR. See the literature.

This chapter is only an introduction to continuous inverse theory, neural networks, and the Radon transform. The references cited at the beginning of the chapter are an excellent place to begin a deeper study of continuous inverse problems. The Backus–Gilbert approach provides one way to handle the construction (i.e., finding a solution) and appraisal (i.e., what do we really know about the solution) phases of a continuous inverse analysis. In this sense, these are tasks that were covered in detail in the chapters on discrete inverse problems. For all the division between continuous and discrete inverse practitioners (continuous inverse types have been known to look down their noses at discrete types saying that they're not doing an *inverse* analysis, but rather parameter estimation; conversely, discrete types sneer and say that continuous applications are too esoteric and don't really work very often anyway), we have shown in this chapter that the goals of constructing (finding) a solution and appraising it are universal between the two approaches. We hope to add more material on these topics Real Soon Now.

References

Claerbout, J. F., *Imaging the Earth's Interior*, 398 pp., Blackwell Scientific Publications, Oxford, UK, 1985.